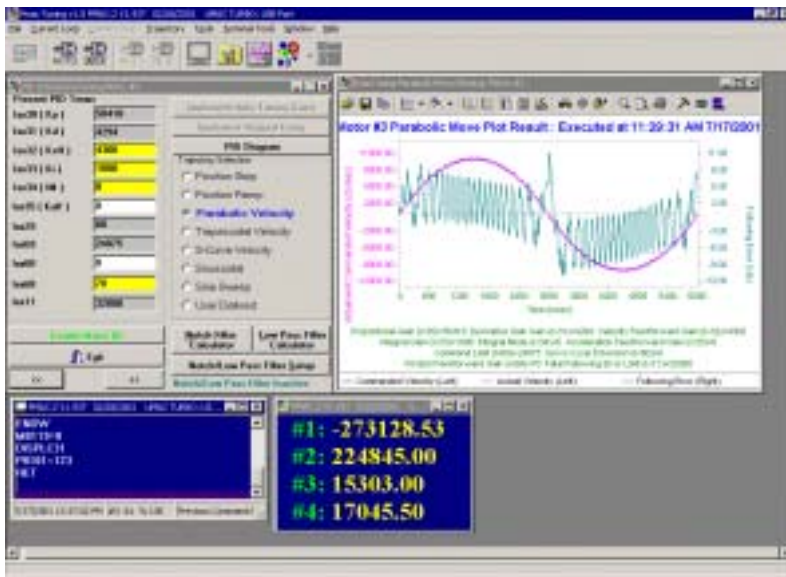


SOFTWARE REFERENCE MANUAL

Pmac Tuning Pro (Release Version 1.x)



Servo Amplifier

3A0-09WPRO-xSx3

January 28, 2003



Single Source Machine Control

21314 Lassen Street Chatsworth, CA 91311 // Tel. (818) 998-2095 Fax. (818) 998-7807 // www.deltatau.com

Power // Flexibility // Ease of Use

License Statement and Limited Warranty
(If you have any questions, contact our Customer Service Department at (818) 998-2095)

IMPORTANT: Carefully read all the terms and conditions of this agreement before installing this software. Installing this software indicates your acceptance of the terms and conditions contained in this agreement. If you do not agree to the terms and conditions contained in this agreement, promptly return this package, unopened, and all associated documentation to the place of purchase, and your money will be refunded. No refunds will be given for products that have missing or damaged components.

LICENSE: This software is protected by the United States copyright law. Therefore, you must treat this software exactly like a book, except that you may copy it onto a computer to be used, and you may make archival copies of the software for the sole purpose of protecting your investment from loss.

By saying, "just like a book," Delta Tau means that this software is free to be used by any number of people and that it may be freely moved from one computer to another so long as it is impossible for it to be used at one location while it also being used at another. Just like a book can not be read by two people in different locations, this software is not permitted to be used by two people in different locations.

Further, you may not download or, by any other means, transmit this software electronically from one computer to another, except as may be specifically allowed in using the software on a computer network (see below). You may transfer all of your rights to use this software to another person, provided that you also transfer all components of this package, including software, diskettes, documentation, and this license agreement, and that you destroy all archival copies and copies installed on a computer for use of this product, including copies in RAM or installed on a disk or diskettes. Remember, once you transfer the software, it may only be used at a single location at a time, as specified in this agreement. At no time may you otherwise transfer, rent, lease, sub-lease, time-share, or lend the software, diskettes, documentation, or any part included herein. Your use of this software is limited to acts that are essential in the use of the software on your computer as described in the documentation. You may not otherwise modify, alter, decompile, or reverse-engineer the software, and you may not remove or obscure Delta Tau's copyright or trademark notices.

NETWORK LICENSE: A "network" is any electronically linked configuration in which two or more users have common access to software or data. Using this software in such a manner is not in accordance with this license agreement, unless you add authorized user licenses as specified below. You may attain additional user licenses by (1) purchasing additional individual software packages for each additional user you wish to add or (2) if a Network Pack is available for this product, paying for the multiple-use license available in the Network Pack. Note that you may use any combination of methods (1) and (2) to increase the number of authorized user licenses.

TERM: This license agreement is in effect until terminated. You may at any time terminate this agreement by destroying the software, diskettes, documentation, and all copies thereof. Delta Tau reserves the right to terminate this agreement if you fail to comply with any of the terms and conditions contained herein. Should Delta Tau terminate this agreement because of your failure to comply, you agree to destroy or return to Delta Tau the program and documentation and any copies, in any and all forms, received from Delta Tau or generated in connection with this agreement.

LIMITED WARRANTY: Delta Tau warrants that the diskettes and documentation enclosed within this product will be free from defects in materials and workmanship for a period of ninety days from the date of purchase as evidenced by a copy of your receipt. **THE PROGRAM IS PROVIDED "AS-IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.** This limited warranty gives you specific legal rights; you may have others which vary from state to state. Some states do not allow the exclusion of incidental or consequential damages so some of the above may not apply to you.

The entire and exclusive liability and remedy for breach of the Limited Warranty shall be limited to replacement of defective diskette(s) or documentation and shall not include or extend to any claim for or right to recover any other damages, including but not limited to, loss of profit, data, or use of the software, or special, incidental, or consequential damages or other similar claims, even if Delta Tau has been specifically advised of the possibility of such damages. In no event will Delta Tau's liability for damages to you or any other person ever exceed the actual original price paid, as evidenced by the receipt, for the license to use the software, regardless of any form of the claim. In the event that the original receipt is lost, the suggested list price at the time of purchase will be substituted as the maximum amount for liability for damages.

GOVERNMENT: This license statement shall be construed, interpreted, and governed by the laws of the State of California. If any provision of this statement is found void or unenforceable, it will not effect the validity of the balance of this statement, which shall remain valid and enforceable according to its terms. If any remedy provided is determined to have failed of its essential purpose, all limitations of liability and exclusions of damages set forth in the Limited Warranty shall remain in full force and effect. This statement may only be modified in writing signed by you and an authorized officer of Delta Tau. Use, duplication, or disclosure by the US Government of the computer software and documentation in this package shall be subject to the restricted rights applicable to commercial computer software. All rights not specifically granted in this statement are reserved by Delta Tau.

Delta Tau Data Systems, Inc. 21314 Lassen Street, Chatsworth CA. 91311 (818) 998-2095

All Delta Tau products are trademarks of Delta Tau. Copyright ©1995, 1996, 1997, 1998, 2001 Delta Tau. All right reserved.

Contents

OVERVIEW	1
INTRODUCTION	1
FEATURES	1
MANUAL LAYOUT	1
<i>How This Manual Is Organized</i>	1
HARDWARE AND SOFTWARE REQUIREMENTS	2
CHAPTER 1 - GETTING STARTED	1
INSTALLING PMACTUNINGPRO	1
SETTING UP COMMUNICATIONS WITH PMAC	1
TECHNICAL SUPPORT	1
CHAPTER 2 - MENU OVERVIEW	3
FILE MENU	3
<i>Select Pmac</i>	3
<i>Open</i>	3
<i>Save</i>	4
<i>Save As</i>	4
<i>Print</i>	4
<i>Exit</i>	4
WINDOW MENU	4
CHAPTER 3 - CURRENT LOOP TUNING	7
PREPARATORY HARDWARE AND I-VARIABLE SET UP	7
OVERVIEW OF PMAC2, TURBO PMAC2 AND UMAC DIGITAL CURRENT LOOP	7
INTERACTIVE CURRENT LOOP TUNING	8
<i>A Typical Interactive Current Loop Tuning Session</i>	9
CURRENT LOOP AUTO TUNING	11
CURRENT LOOP AUTO TUNING DIALOG BOX	12
<i>A Typical Automatic Current Loop Tuning Session:</i>	13
CHAPTER 4 - POSITION LOOP TUNING	15
DAC CALIBRATION	15
<i>Number of test Iterations</i>	15
<i>Begin Calibration</i>	15
<i>Graphical offset illustration</i>	16
OPEN LOOP TEST	16
<i>Open Loop Magnitude (%)</i>	18
<i>Open Loop Time (ms)</i>	18
<i>Number of Repetitions</i>	18
<i>Plot Response</i>	18
REGULAR PID TUNING	19
<i>Regular PID Interactive Tuning</i>	19
<i>Present PID Terms</i>	20
<i>Regular PID Auto Tuning</i>	21
<i>Choices & Options in Auto-Tuning:</i>	23
<i>Doing an Auto-Tune Session:</i>	25
EXTENDED SERVO ALGORITHM	26
<i>ESA Interactive Tuning</i>	27
<i>ESA Structure</i>	28
<i>Structure of ESA Diagram</i>	29
<i>ESA Auto Tuning</i>	30
<i>Choices & Options in ESA Auto-Tuning:</i>	33

<i>Doing an ESA Auto-Tune Session:</i>	34
FILTER	36
<i>Notch Filter (Addition to PID + Feedforward)</i>	36
<i>Low Pass Filter (Addition to PID + Feedforward)</i>	37
GANTRY PID.....	39
<i>Gantry Auto Tuning</i>	39
<i>Gantry Interactive Tuning</i>	39
CHAPTER 5 - TRAJECTORIES	41
POSITION STEP	42
<i>Doing the Step Response:</i>	43
<i>Initial step move response</i>	44
<i>Modified step response with higher Kd (Ix31)</i>	45
<i>Modified step response with higher Kp (Ix30)</i>	46
<i>Modified step response with increased Kd again</i>	47
<i>Modified step response with increased Ki (Ix33)</i>	48
<i>Modified step response with very high Ki</i>	49
POSITION RAMP	50
PARABOLIC VELOCITY	51
<i>Doing the Parabolic Move:</i>	52
<i>Modified parabolic move with increased Kvff (Ix32)</i>	53
<i>Modified parabolic move with Ix34 = 1</i>	54
<i>Modified parabolic move with too much Kvff</i>	55
<i>Modified parabolic move with some Kaff</i>	56
TRAPEZOIDAL VELOCITY	56
S-CURVE VELOCITY	57
SINUSOIDAL	58
SINE SWEEP	59
USER DEFINED	60
CHAPTER 6 - TOOLS.....	61
CONFIGURATION	61
TERMINAL	62
STRIP PLOT	62
REPORT	64
MOTOR STATUS	65
POSITION	66
CHAPTER 7 - HELP	67
PMAC TUNING HELP	67
ABOUT	67

OVERVIEW

Introduction

Welcome to the PmacTuningPro, Delta Tau's tuning utility. The PmacTuningPro is part of the Delta Tau software suite. PmacTuningPro enables you to configure and improve the different control loops in your PMAC(s).

PmacTuningPro is a configuration tool as well as a diagnostic tool to help create and manage the various aspects of tuning your motors. With PmacTuningPro you will be able to configure and optimize your digital current loop gains; verify the open loop operation of your motor; calibrate servo command offsets; configure and optimize position loop gains, for standard PID loop as well as Extended Servo configurations, for single motors and gantry systems, and implement various position and velocity filters. You can test and optimize your position and velocity loop configuration using various trajectories. In addition to the tuning functions, PmacTuningPro provides real-time plotting utilities, motor status screen, watch window, position window and a simple terminal to help monitor and perform different tasks on your system.

Features

PmacTuningPro provides basic tools for Current Loop Tuning, PID Tuning, Extended Servo Algorithm Tuning, Trajectories, DAC Calibration, Open Loop Test, Notch Filter and Low Pass Filter. Real-time motor status display.

Manual Layout

This manual explains how to use PmacTuningPro to communicate with and Tune motors using your PMAC motion control card. Knowledge of the basic use of the Windows operating system is assumed.

How This Manual Is Organized

- **Chapter 1 - Getting Started.** Covers installing the software and setting up communication with PMAC. Provides Delta Tau support information.
- **Chapter 2 - Menu Overview.** Gives a brief description of each of PmacTuningPro's menus.
- **Chapter 3 - Current Loop Tuning.** Provides a quick tool for tuning of the Current Loop. This is a repetition of initial setup of the Current Loop Tuning using the P2setup program (for PMAC2) or the TurboSetup program (for PMAC2 Turbo and UMAC).
- **Chapter 4 - Position Loop Tuning.** Details all of the PID and Extended Servo Algorithms steps including DAC calibration, Open Loop test, Filters and Gantry Tuning.
- **Chapter 5 – Trajectories.** Provides several pre-defined as well as custom-defined moves.
- **Chapter 6 – Tools.** Allows viewing several diagnostic and static tools, including Terminal windows, Position and status screens.
- **Chapter 7 – Help.** Details PmacTuningPro's help information routines.

Hardware and Software Requirements

PmacTuningPro will run on any computer capable of running Windows 98/ME or 2000 (TM) (266 MHz Pentium MMX and up recommended). In addition, you will need the following:

Microsoft Windows 98/ME or Windows 2000 loaded on your computer.

At least 16 MB of free disk space and 32-48 MB of RAM (32 MB for Windows 98/ME or 48 MB for Windows 2000).

A free serial communications port, or USB port, or Ethernet port, or PCI-BUS slot, or ISA-BUS slot to talk to PMAC for on-line processing.

Any monitor with VGA resolution (800x600 suggested but 640x480 works fine).

CHAPTER 1 - GETTING STARTED

Installing PmacTuningPro

Before installing PmacTuningPro, read the license agreement included in this manual (behind title page), and make a backup copy of the installation disks. To install PmacTuningPro from the Delta Tau Software CD, insert the CD into the CD drive. Auto install menus will popup. Click on the PmacTuningPro from the Suite to launch PmacTuningPro installation. To install PmacTuningPro from the floppy disks, put the PmacTuningPro distribution disk labeled "Disk #1" into a floppy drive and choose File | Run from the Program Manager. Enter A:\SETUP.EXE or substitute 'A' for the letter of your floppy drive.

PComm32 drivers were rewritten to accommodate the addition of PCI, USB and Ethernet communication modes. It is therefore important to uninstall all old Delta Tau software products before installing PmacTuningPro.

The installation program will suggest a directory path where the program files should be copied. Please use the suggested directory location for the installation for the purposes of uniformity among all PmacTuningPro users (and trouble shooting if need be).

Read the "readme32.txt" file for last minute additions to this manual.

Setting up Communications with PMAC

No applications, including PmacTuningPro, will be used to add, remove or configure PMACs in your system. Rather, communication settings have been centralized in your operating system, making the set up of each PMAC much like other devices in your computer (i.e. video card, sound card etc.) All setup is done through the Control Panel's "Add New Hardware" wizard. Installation steps as provided in the *Pcomm32PRO Install* manual will help installing and registering the newly installed devices. Before running this application it is important that all applications that use PComm32 (the Delta Tau 32-bit communication driver) be shut down. This includes PmacTuningPro, Pewin32PRO, NC for Windows, and any applications developed with PComm32 or PTalkDT.

Technical Support

Delta Tau is happy to respond to any questions or concerns you have regarding the Windows Executive. By far, you'll get the quickest response if you send your queries to the following e-mail address:

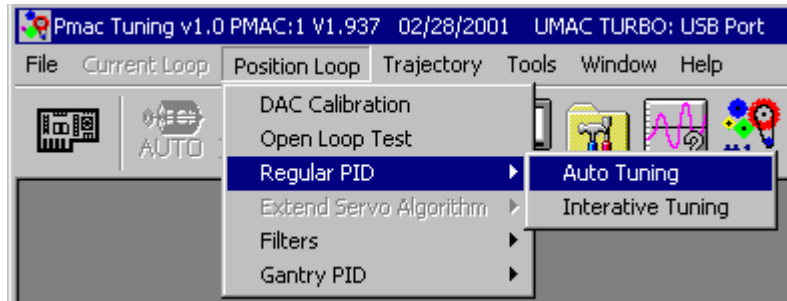
support@deltatau.com. Of course please do check out our Web site at WWW.DELTATAU.COM.

You can call Delta Tau Monday through Friday from 9:00 AM to 4:30 PM PST or FAX us your request or problem, and we will deal with it the next business day.

Delta Tau Data Systems, Inc. 21314 Lassen Street Chatsworth CA, 91311	West Coast	East Coast
	Voice : (818) 998-2095	Voice : (804) 795-4288
	FAX : (818) 998-7807	FAX : (804) 795-4996

CHAPTER 2 - MENU OVERVIEW

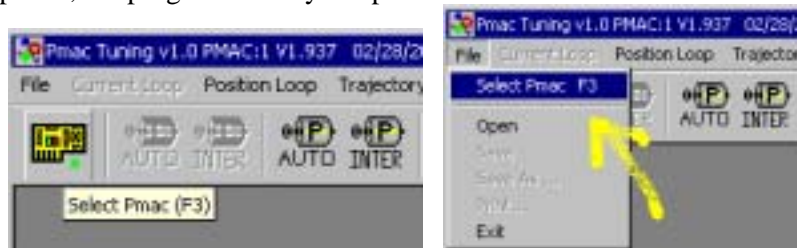
This chapter describes the File menu and the window menu. All other menu items are explained in the respective chapters.



File Menu

Select Pmac

Choose which PMAC card you want to tune, or to test the terminal window. The terminal window will pop up automatically. When the program closes, the device number is written to INI file. The next time PmacTuningPro is opened, the program will try to open that device number.



Open

Opens Pmac-type file like Ivar (*.ivr) and Pmac file (*.pmc) and all text type file.



Save

Saves any open text file without changing the filename.



Save As

Saves any open text file, with a new filename.



Print

Prints any open text file to the printer.



Exit

Saves configuration to INI file, and exits the program.



Window Menu

This menu is for managing the position and arrangement of all windows currently displayed.

<u>W</u> indow	<u>H</u> elp
Cascade	
Tile Horizontally	
Tile Vertically	
Minimize All	
Arrange All	
1 Motor Setup Summary	
2 Strip Plotting: Motor #1	
✓ 3 PMAC:0 V1.937 01/16/2001 PMAC1 TURBO: ISA BUS	

CHAPTER 3 - CURRENT LOOP TUNING

(Digital Current Loop Tuning for PMAC2, Turbo PMAC2 and UMAC)

This feature has been added to PMAC Tuning Pro to provide a quick tool in case re-tuning of the Current Loop is needed after the initial setup of the Current Loop using the P2setup program (for PMAC2) or the TurboSetup program (for PMAC2 Turbo and UMAC.)

Preparatory Hardware and I-variable Set Up

When PMAC2's digital current loop is used, the format of the output commands is virtually always digital pulse-width-modulated (PWM) signals. For each machine interface channel, PMAC2 has 3 pairs of top and bottom PWM signals, which can be used for the half bridges of a 3-phase motor. Each of the 6 output signals is a differential line-driver pair, for a total of 12 PWM pins which must be correctly connected to the amplifier. The analog current feedback signals from two of the motor phase windings are also required and must be correctly connected to PMAC2 (for more details see the appropriate sections of Delta Tau Digital Amplifier manual and the PMAC2 User Manual).

In addition to the need for correct inter-connection wiring between the PMAC2 and the Digital Amplifier, some global, per channel and per motor I-variables must be preset prior to any attempt to tune the current loops. These I-variables include the global variables:

- I900 (I7m00 for Turbo) and I906 for PWM frequencies,
- I901 (I7m01 for Turbo) for Phase Clock frequency,
- I902 (I7m02 for Turbo) for Position loop Servo Clock frequency,
- I903 (I7m03 for Turbo) and I907 for the ADC Clock frequencies,
- I904 (I7m04 for Turbo) and I908 for the PWM deadtime between top and bottom signals for machine interface channels 1 to 8.

The factory default values for the per-channel PWM I-variables I9n6 (I7mn6 for Turbo) and I9n7 should also be inspected and, if necessary, modified accordingly. The relevant per-motor I-variables are Ix00, Ix01, Ix02, Ix66 and Ix82, which should all be set up correctly.

Overview of PMAC2, Turbo PMAC2 and UMAC Digital Current Loop

The "field oriented" electronic commutation algorithm used on PMAC2 requires the closure of two feedback loops around the motor current vector in the "field frame." This algorithm transforms the digitized values of the motor phase currents to a current vector in the field frame and then resolves the vector into two components: the torque producing *quadrature* current I_q , and the field producing *direct* (magnetizing) current I_d . Each component of the feedback signal is independently utilized for the regulation of I_q and I_d using a *Proportional plus Integral* (PI) control algorithm. The pertinent motor I-variables for current loop tuning are:

Ix61..... the current loop integral gain K_i

Ix62..... the current loop forward path proportional gain K_{p2}

Ix76..... the current loop feedback path proportional gain K_{p1}

Ix66..... the PWM Scale factor

Ix66, the PWM Scale Factor, must be set correctly according to the chosen PWM frequency (i.e. Ix66 should be set to a value slightly greater than I900 or I906). Once Ix66 is set to a reasonable value it should not be altered, as any modification of this variable will affect the loop's dynamic response. To tune or re-tune the current loops, one should only alter the values of Ix61, Ix62 and/or Ix76.

Whenever a motor is being electronically commutated using the PMAC2's "field oriented" commutation algorithm, one of the first steps would be to tune the current loops. A stable and responsive current loop is a

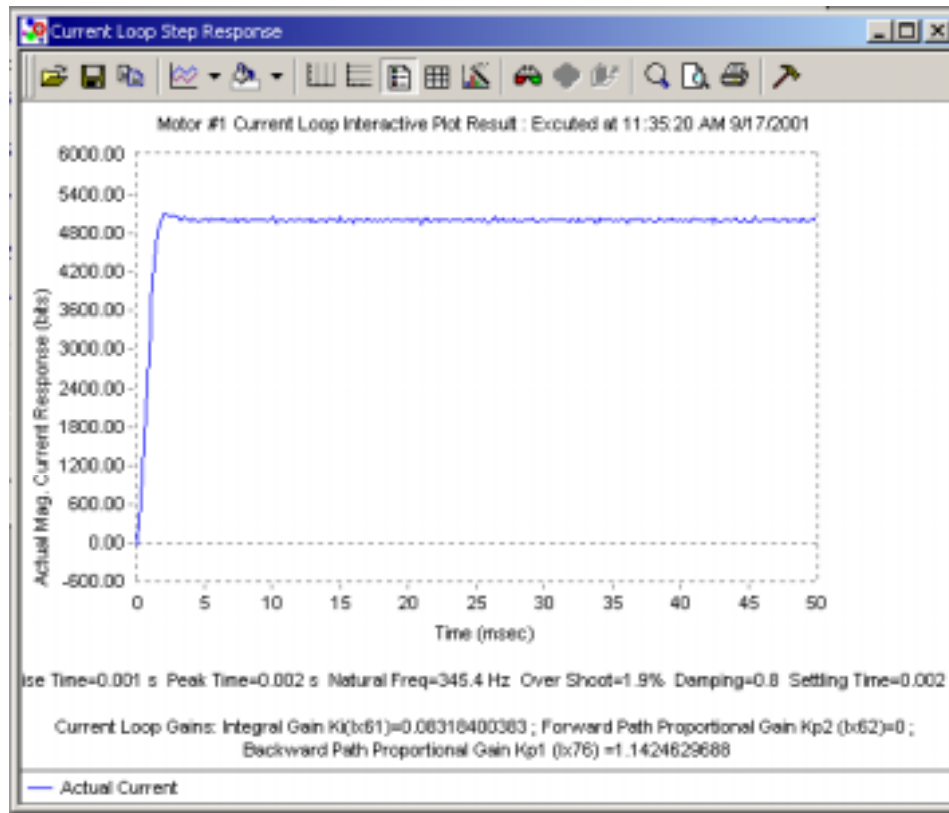
prerequisite for a satisfactory motor torque control and subsequently motor velocity and position control. The process of tuning the current loop is usually carried out prior to setting up the PMAC2 commutation I-variables (i.e. Ix70.. Ix79). It is therefore important to note that the correct setup of the PMAC2's commutation I-variables (Ix70..Ix 79 and Ix83) is not required for current loop tuning. PmacTuningPro has two current loop tuning aids: the Interactive Tuning and the Auto Tuning options. Each option is discussed below.

Interactive Current Loop Tuning



To tune the current loop, adjust the gains Ix61, Ix62 and/or Ix76 and then execute a step current command. This "step" in current command is in fact a step in the *direct* or the magnetizing current, which should produce little or no torque and hence a very small amount of motion (if any). A time trace (as shown in the example below) provides numerical values of the pertinent characters of the step response including the *natural frequency* and *damping ratio*. As the integral gain Ix61 is increased, the natural frequency will be increased, manifested by a faster rise time. However, the dampness characteristic of the response will be reduced. This is shown by longer settling time and increased % overshoot. To provide the required damping

(typically a damping ratio between 0.7 to 1.0), either or both of the proportional gains Ix62 and Ix76 may be increased. The iterative process of increasing the integral gain to produce faster rise time (higher *natural frequency*) and increasing the proportional gain(s) to produce less overshoot and faster settling (*damping ratio* between 0.7 to 1.0) could be continued until the closed loop current response exhibits either no further rise time improvements (due to saturation) or noise induced "buzzing" (due to ADC quantization of the current signal). In either case one must reduce the gains and test for a satisfactory behavior via the step response test again.



A Typical Current Loop Step Response

In summary, increasing the integral gain Ix61 increases the natural frequency (speed of response) but also reduces the damping ratio (increases the settling time). To compensate for the reduction in the damping ratio, one should increase either or both of the proportional gains (Ix62 and/or Ix76). The choice between the two proportional gains is often dictated by a trade-off: the noisiness of the current sensor versus the resolution of the velocity sensor (pointed to by Ix04). For systems with low resolution velocity sensors, it is best to use Ix76 (the feedback path proportional gain) instead of Ix62 (the forward path proportional gain). For systems with noisy current sensors, it is best to use Ix62 instead of Ix76. In general, for a given numerical value of the proportional gain, when it is implemented through Ix62, a faster response is achieved. However, the response may have somewhat more overshoot.

A Typical Interactive Current Loop Tuning Session

Prior to any attempt to use the Interactive Current Loop tuning option, make sure that the pertinent I-variables (see above) have been set up and that the PWM output, the current feedback signals and, if necessary, Amplifier Enable signal are all correctly connected between the motor and the Digital Amplifier.

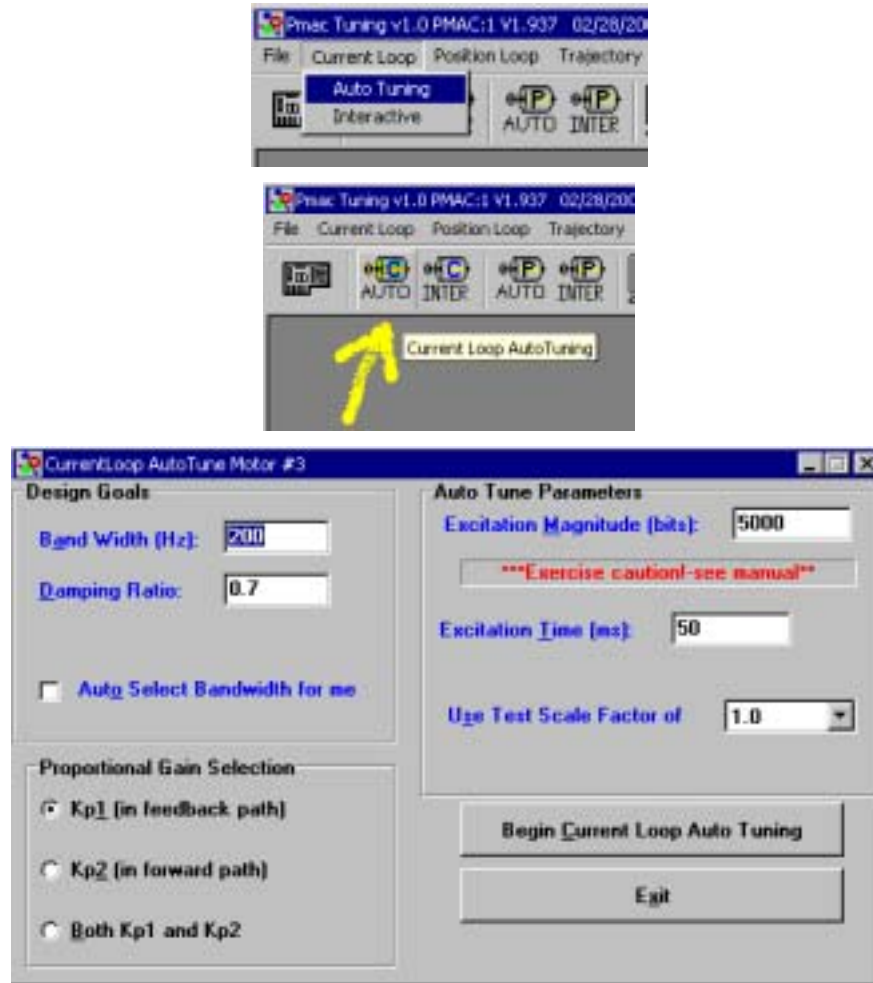
WARNING

It is both unnecessary and possibly dangerous to have any mechanical load connected to the motor during a current loop interactive tuning session.

During the execution of the current loop step, the Executive program has to increase the servo frequency to be the same as the phase frequency for data gathering purposes. In addition, all other motor channels are disabled momentarily (I_{x00} is set to zero for all motors other than the one being tested) until the end of the step test. **It is therefore most important that no load is connected to any of the motors being controlled by a PMAC2 while any other motor's current loop is being tuned**

Select a **Step Size** equal to the size of the magnetizing current I-variable I_{x77} used in the test. The size of the step should be sufficiently large to overcome any possible amplifier deadband and other low amplitude non-linearity and noise. On the other hand, it should not be too large to avoid any possible inadvertent motion of the motor due to incorrect setting of the commutation parameters. A **Step Size** of 5000 bits is usually a good starting point. A **Step Time** of between 50 to 100 milliseconds should, in most cases, be sufficiently long for this test. By clicking on the **Current Loop Step** button, a step stimulus is injected in the direct (I_d) component of the current loop; the response data is then gathered within PMAC2 and it is plotted on the screen. Since the non-torque producing component of the current vector is stimulated, you should notice almost no motion or a very small amount of motion. If the commutation I-variables were set correctly, no motion should be detected. The numerical value of the natural frequency should be noted from the step response screen. The general aim is to increase it to as high a number as possible while keeping the damping ratio between 0.7 to 1.0 (see above). Numerically, you should aim for a natural frequency of at least 150 to 200 Hz for a motor with a high winding impedance. For a motor with a small winding impedance the attainable value of natural frequency may be as high as 400 to 500 Hz. In general, the current loop should be tuned to have a natural frequency that is numerically 10 to 15 times higher than the desired position loop's natural frequency. Furthermore, a current loop natural frequency of less than 100 Hz is usually not acceptable for a smooth (low cog) torque generation, even when the required position loop natural frequency is below 10 Hz. No attempt should be made to tune the position loop until the current loop tuning is completed, the commutation I-variables are set up, and the mechanical load is connected to the motor.

Current Loop Auto Tuning



The objective of the current loop Auto Tuner is to automatically set up the PI gains according to the desired values of natural frequency (bandwidth) and damping ratio. The Auto Tuner carries out a number tests with a step stimulus fed into the direct (magnetizing) current loop. During these tests the integral gain Ix61 is not used, and the proportional gain Ix76 is set to a fixed value. Based upon the result of the tests, the program attempts to estimate the impedance of the motor winding and subsequently it calculates the required PI gains to achieve the desired natural frequency (bandwidth) and damping ratio.

Current Loop Auto Tuning Dialog Box

1. **Bandwidth-** This parameter specifies the desired *natural frequency* of the current loop. The specified desired current loop bandwidth should be at least 10 to 15 times the desired position loop bandwidth (the position control loop should be tuned once the current loop tuning is completed and the commutation I-variables are correctly set and the load is connected to the motor). Furthermore, **a current loop natural frequency (bandwidth) of less than 100 Hz is usually not acceptable for a smooth (low cog) torque generation.** The attainable bandwidth does depend on the amplifier and the motor phase winding impedance. For low impedance motors, bandwidths as high as 500 Hz (or more) may be achievable without saturating the gains. For high impedance motors you may experience saturation limits if the desired bandwidth is set too high. Bandwidths in the range of 200 to 500 Hz are usually acceptable for most applications.
2. **Damping Ratio-** This parameter specifies the desired damping ratio, which is a measure of how fast the current loop transients settle down or dampens out. A suitable numerical value for the desired damping ratio is between 0.7 and 1.0.
3. **Auto-Select Bandwidth-** When this box is checked, the Auto Tuning algorithm automatically selects a conservative (lower bound) bandwidth, replacing the **Bandwidth** entry above. This option is useful for providing a shortcut in the convergence process to the maximum achievable current loop bandwidth for a given amplifier/motor combination. When this box is checked, the Auto Tuner uses internal calculations on the results of the step stimulus test(s) to determine a conservative bandwidth. The recommended procedure is to check this box and carry out an auto tuning cycle, noting the automatically calculated bandwidth displayed in the **Bandwidth** entry. Then uncheck this box and gradually increase the **Bandwidth** entry. Carry out another auto tuning cycle until one of the gains saturates. The final selected gains should be those that provide the highest bandwidth without causing saturation.
4. **Maximum Excitation Magnitude-** This is the *maximum* step size of the commanded *direct* (magnetizing) current I_d used within the identification tests (the numerical range of this entry is the same as I_{x77}). For the first iteration of the identification process, this number is divided by the **Number of Iterations** to form the first step size. The last iteration has a step size equal to the **Max Excitation magnitude**. Select this entry very carefully. If the **Max Excitation magnitude** is selected to be too large, it may cause sudden motion of the motor or possibly damage the amplifier. If the numerical value of **Max Excitation magnitude** is set to be too small the current feedback noise-to-signal ratio may be unacceptably high for the correct identification of the motor winding impedance. A numerical value of 5000 is usually a good starting point. If you observe too much noise-to-signal ratio in the step response, progressively increase this number.
5. **Excitation Time-** This is the period of time for the *direct* (I_d) current step test. A period of 50 ms should be sufficiently long to achieve steady state values. In general, this period should be made as short as possible and yet long enough for the current response to reach its steady state.
6. **Number of Iterations-** This is the number of step stimulus tests during an auto tuning cycle. If a number greater than 1 is chosen, then the commanded step size at the first iteration will be the **Maximum Excitation Magnitude** entry divided by the **Number of Iterations** entry. In most cases, one or two iterations should be sufficient for a successful identification of the current loop parameters.
7. **Plot Current Curves-** When this option is checked the results of the direct current test stimuli for each iteration step are shown graphically. Note that, since all the test stimuli use only the proportional gain, the steady state response is likely to show a value that is smaller than the desired step.
8. **Kp1 (in feedback Path)-** When this radio button is selected, the required proportional gain determined by the Auto Tuner will be implemented in the feedback path through the use of I_{x76} . Choose this option if you have a low noise current feedback signal. This implementation of the

proportional gain in the feedback path helps reduce the quantization noise effects from the position sensor. It is particularly useful when low resolution position sensors are used.

9. **Kp2 (in forward Path)-** When this radio button is selected, the required proportional gain determined by the Auto Tuner will be implemented in the forward path through the use of Ix62. Choose this option if you observe excessive noise in the current feedback signal. This implementation of the proportional gain in the forward path may amplify the quantization noise effects from the position sensor. Avoid this option when low resolution position sensors are used.
10. **Kp1 and Kp2-** When this radio button is selected, the required proportional gain determined by the Auto Tuner will be implemented partially in the forward path and partially through the feedback path. Here, an attempt is made for the trade-off between a low-resolution position feedback signal and a noisy current-feedback signal in selecting these gains.

A Typical Automatic Current Loop Tuning Session:

Prior to any attempt to use the Auto Tuner for the Current Loop, make sure that the pertinent I-variables (see above) have been set up and that the PWM output and the current feedback signals and, if necessary, Amplifier Enable signal are all correctly connected between the motor and the Digital Amplifier.

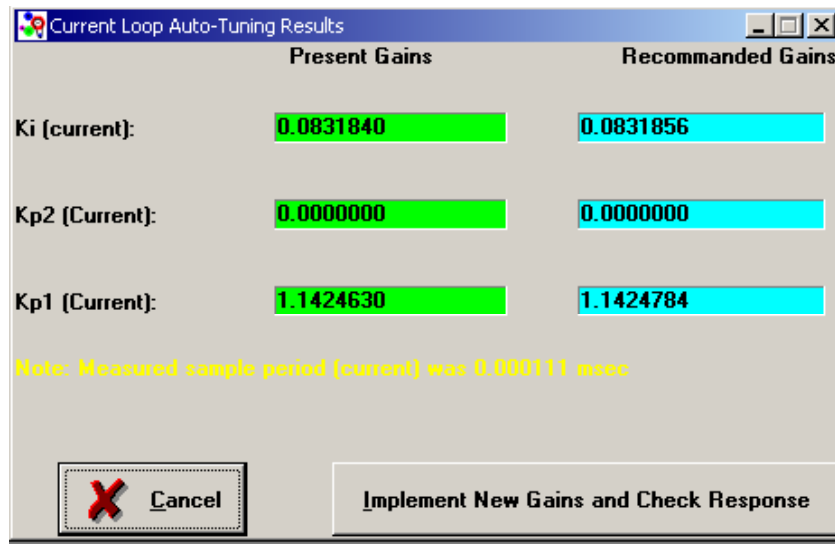
WARNING

It is both unnecessary and possibly dangerous to have any mechanical load connected to the motor during a current loop auto tuning.

During the execution of the auto tuning tests, the Executive program has to increase the servo frequency to be the same as the phase frequency for data gathering purposes. In addition, all other motor channels are disabled momentarily (Ix00 is set to zero for all motors other than the one being tested) until the end of the step test. **It is therefore very important that no load is connected to any motor being controlled by a PMAC2 while any other motor's current loop is being auto tuned.**

1. Make sure that no load is connected to any of the motors being controlled by the PMAC2 board, and that the motor/amplifier connections and PMAC2/amplifier connections are correctly wired.
2. Enter the Executive program Current loop Auto Tuning option and select the desired motor number to be auto tuned.
3. Click on the **Check Current Loop Polarity** option. If the result is positive feedback (incorrect current feedback polarity) stop. Reverse the current feedback signal polarity and check again for the correct (negative feedback) sense before proceeding further.
4. Check the **Auto-Select Bandwidth** option and select a safe **Maximum Excitation Magnitude** value (see above). You may leave the other options at their default values. A **Damping Ratio** entry between 0.7 to 1.0 should be used throughout the auto tuning activities. Check the **Plot Current Curves** option and click on the **Begin Current Loop Auto Tuning** button. You should see (depending on the selected number of iterations) a number of step response curves. During these tests the integral gain is set to zero, which means that you will see a steady state error between the desired and the actual step size. This is intentional. Finally a message

box will appear, giving you the calculated gains based on the bandwidth that the Auto-Tuner has selected. If during the auto tuning cycle, the responses indicate excessive noise, you may wish to click on the **Cancel** button. Try a larger value for the **Maximum Excitation Magnitude** entry and repeat the auto tuning cycle. When you are satisfied with the signal-to-noise ratio of the step response you may click on the **Implement and Check Response** button. You should see a step response that shows a numerical value for the *natural frequency*. This number should be very close to the *bandwidth* selected by the Auto Tuner. If it is to within 10 to 20% of the selected *bandwidth* (in Hz) you may consider that the Auto Tuner has a fairly good estimate of your current loop dynamics.



A Typical Current Loop Auto Tune Message Box

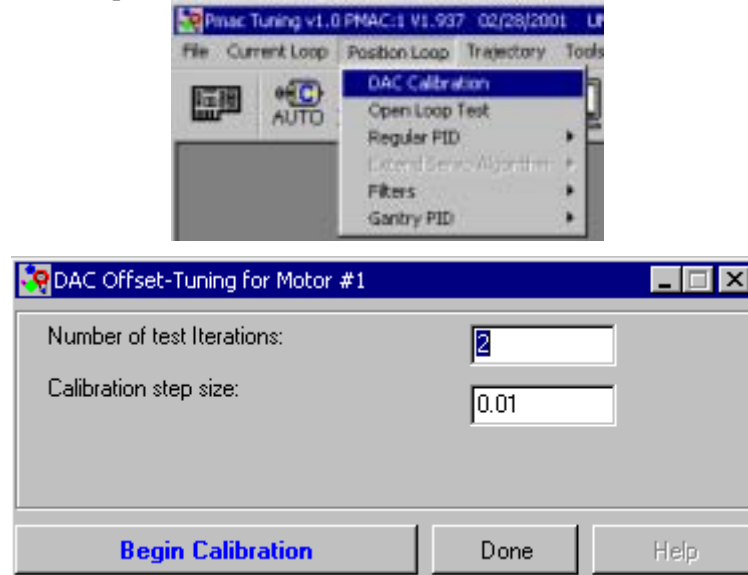
- Now uncheck the Auto-Select Bandwidth option and gradually increase the Bandwidth entry and repeat the above procedure until a message appears indicating saturation of the gains. At that time reduce bandwidth entry and try to converge to the highest bandwidth that just avoids causing saturation. This will be your highest achievable bandwidth using the Auto Tuner with the proportional gain in the feedback path (Kp1). If a higher value of bandwidth is desired you may try to use the proportional gain in the forward path (Kp2).

Once a satisfactory current loop response is achieved for all motors, the current loop gains, SAVE within PMAC2. The next step is usually the setup procedure for PMAC2's commutation I-variables, and finally position loop tuning.

CHAPTER 4 - POSITION LOOP TUNING

DAC Calibration

The purpose of this option is to provide the ability to determine the DAC bias value(s).



For a standard DC motor, or a brushless motor which is commutated externally (not by PMAC), this option provides an optimum value for Ix29. For a stepper motor, driven through PMAC's Accessory 8D Option 2 (the V/F board), this option provides optimum values for Ix29 and Ix79. When you begin calibration, a small PLC program is downloaded and executed in PMAC. After the tests are completed, the optimum values for Ix29 and Ix79 (if you specified a stepper motor) are displayed. Select IMPLEMENT NOW to implement these bias gains in PMAC. Do not use this option if PMAC is commutating your selected motor! See also the special application note included with PMAC ACC-8D Opt. 2 manual, PMAC Setup For Stepper Motor Control Using ACC-8D Opt. 2, The V/F Converter.

WARNING

Regardless of the state of the selected motor's servo loop (close or open), it will be opened up by the Offset Calibration routines. In a situation where the DAC bias voltage is known to be high, the fact that the servo loop is open may lead to substantial motion during the Offset Calibration procedure. Therefore make sure that the motor is free to rotate an unknown number of revolutions during the calibration.

Number of test Iterations

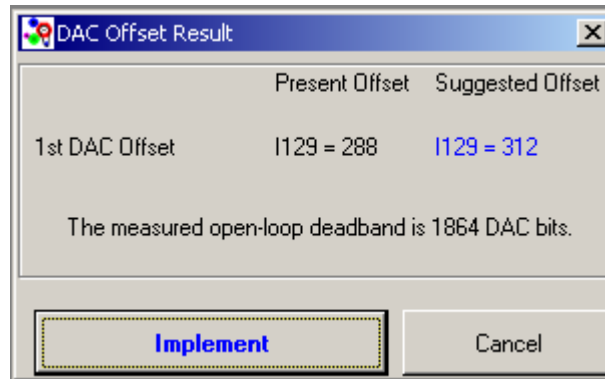
This is the number of backward and forward open loop motions in the testing cycle. Value values are from 1 to 10, and the default value is 4. The calibration program automatically averages the results.

Begin Calibration

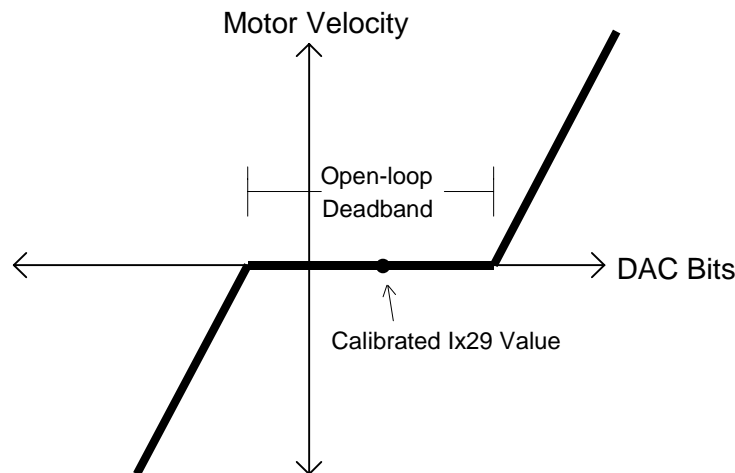
Selecting this button begins the calibrating procedure. Depending on the number of iterations you selected and your motor's friction and open-loop deadband characteristics, the time necessary to complete offset calibration will vary. What the procedure does is increment the DAC in small, positive steps until positive

motion is detected. The DAC is then decreased in small, negative steps until negative motion is detected. This completes one iteration.

After the test is complete, the results of the calibration are presented, showing the current and suggested values for Ix29, and the open-loop deadband size, in encoder counts. This quantity is not the programmable PMAC deadband set by Ix65, but rather the hardware (or open-loop) deadband resulting from the imperfections in the amplifier/motor/load combination. This quantity equates to twice the number of DAC bits, that need to be added to or subtracted from Ix29 to provide a noticeable motion in either direction. Due to amplifier deadband and/or friction in the drive, the value of this quantity is often non-zero. However, it should be a small number if a responsive closed-loop performance is desired. If a large deadband is reported (say larger than 500 bits) there may be a potential for imperfect servo performance, particularly when the gains are low. However, do not set Ix29 equal to the deadband. The value of Ix29 should be set equal to the offset value.

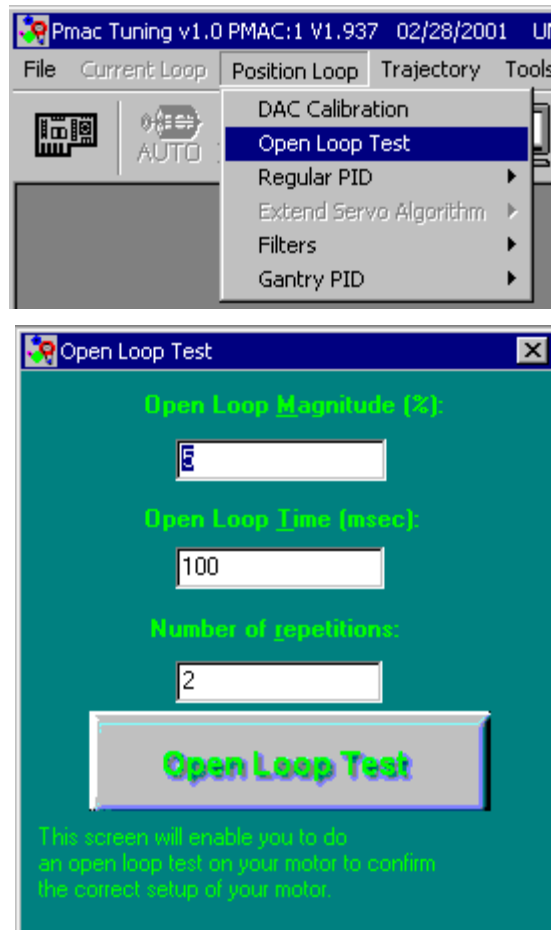


Graphical offset illustration.



Open Loop Test

This button opens a dialog box that allows you to perform a repetitive sequence of open-loop moves according to the values set in Open Loop Magnitude, Open Loop Time, and Number of Repetitions. The open loop move will be performed when you press the OPEN LOOP TEST button. press <x> to close this screen.



This section of the program allows you to program a repetitive sequence of open-loop outputs from PMAC, using its O command feature. This is very useful for tuning amplifiers, particularly those with tachometer feedback. It can take the place of special hardware devices that perform the same purpose. In addition, it has safety features that no hardware device can have: automatic shutdown on exceeding (programmable) velocity and position limits. It also uses data gathering and plotting to display the results of the stimulation (velocity, position, and acceleration versus time), eliminating in many cases the need for an oscilloscope to do the tuning.

You will need to set the open loop magnitude, which is expressed as a percentage (0 - 100%) of the value stored in Ix69. Ix69 holds the maximum allowable output magnitude (in binary bits) which can range from 0 to 32767 corresponding to 0 to 10 volts. The factory default for Ix69 is 20480 (which is about 6.3 volts), but many users reset this to the full 32767. The open loop time specifies the time duration of the "on state" (when a positive or negative value is written to the DAC output) for the open loop move. The Open loop zero time specifies the time duration of the "off state" (when a value of zero [0] is written to the DAC output) for the open loop move. No of reps specifies the number of repetitions for the open loop move cycle. A value of zero (0) indicates infinite repetitions until the space bar is pressed.

When the open loop program is executed, the moving motor's velocity will be limited to the velocity limit as specified by Ix16. This is intended to protect the system by preventing it from moving too fast. This feature depends on having proper encoder feedback to the card. The open loop program is terminated if the velocity limit is exceeded in either the positive or negative direction. In addition, if either the hardware or software position limits are exceeded, the open loop program will stop. The hardware limits are the external limit switches connected to PMAC. The software limits are defined by Ix13 and Ix14 for the positive and negative position limits, respectively.

Open Loop Magnitude (%)

Here you specify the magnitude (maximum voltage in one direction) for the open loop move. This is expressed as a percentage (0 - 100%) of the value stored in Ix69 which holds the maximum allowable output magnitude (in DAC bits). This can range from 0 to 32767, corresponding to 0 to $\pm 10V$. The factory default for Ix69 is 20480 (which is about $\pm 6.3V$), but this is usually reset to the full 32767 ($\pm 10V$).

Open Loop Time (ms)

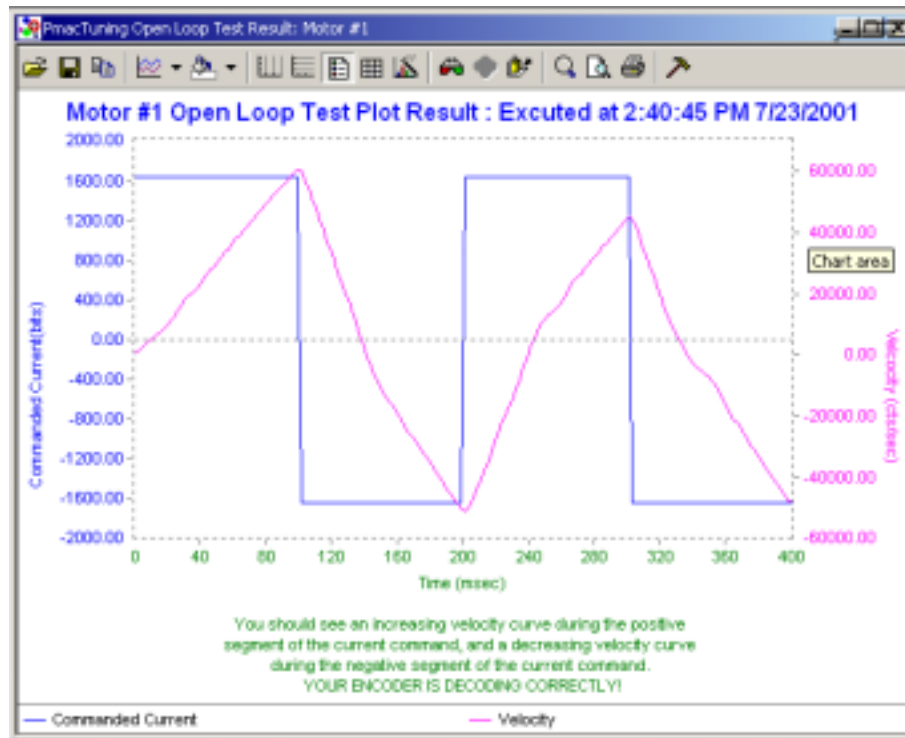
The open loop time specifies the time duration of the "on time" (when a positive or negative value is written to the DAC output) for the open loop move.

Number of Repetitions

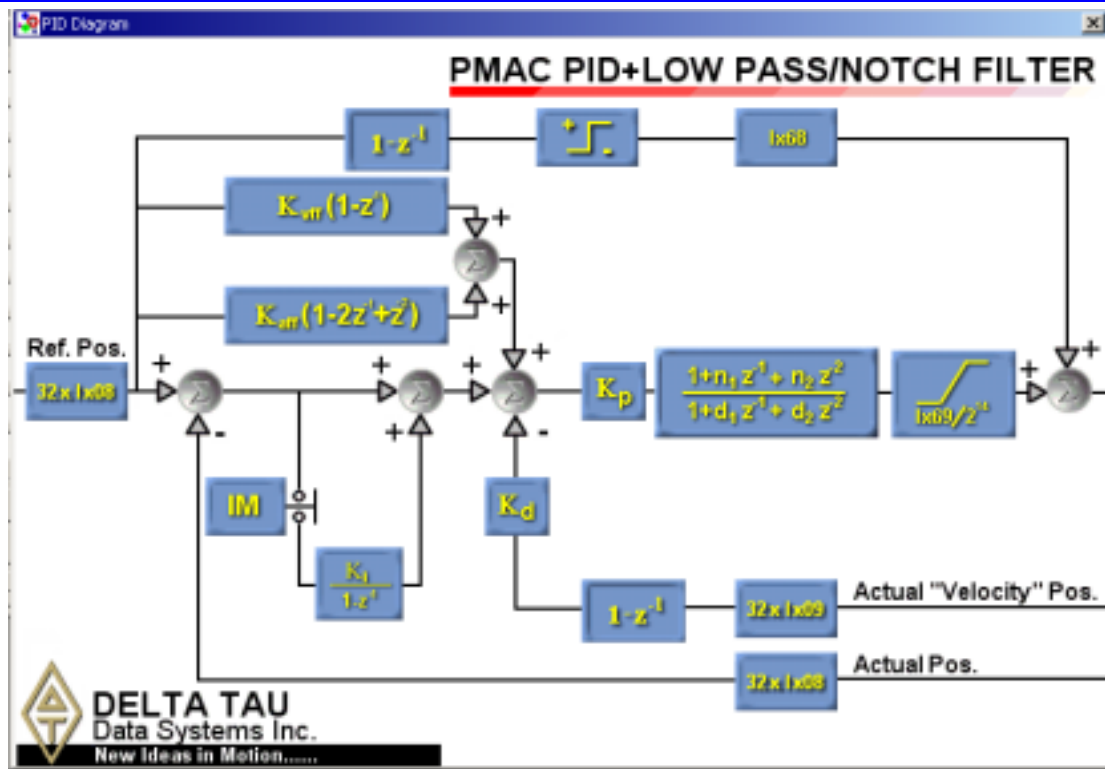
This specifies the number of repetitions for the open loop move. A value of zero indicates infinite repetitions until the space bar is pressed.

Plot Response

This option lets you view the last response plot without redoing the move.



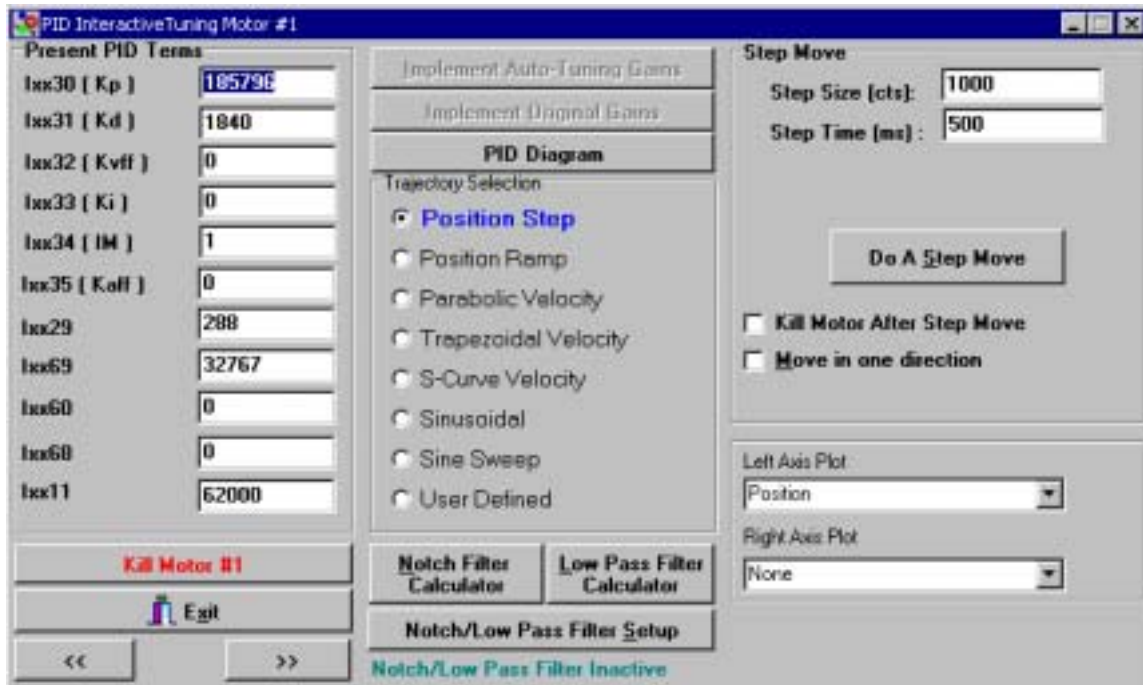
Regular PID Tuning



Regular PID Interactive Tuning

Using interactive Position Loop Tuning in Pmac Tuning should ideally be only for fine tuning, after automatic tuning generates the “ball park figure” for required gain settings. In this section simplified explanation of each gain and parameters will be done so that you can make intelligent guesses.





Present PID Terms

Ixx30(Kp) - Proportional Gain - Increasing proportional gain stiffens the servo loop and increases the natural frequency of the closed loop system. Theoretically, increasing the proportional gain will result in improved positioning and tracking. But often for real systems increasing the proportional gain increases their sensitivity to the noise and disturbances. If you are skipping the auto tuning, begin tuning with the low proportional gain setting. The default value is a very conservative value for most systems, and is a good starting point.

Ixx31(Kd) - Derivative Gain - Derivative gain works like damper. Higher the derivative gain, higher the damping action. This gain prevents overshoot but makes the system sluggish. Also, in digital system the quantization noise is amplified when derivative gain is applied, and in slow moving (low counts per second) system this noise might contribute significantly to the error. If the user is skipping the auto tuning, it is always wise to begin the tuning with the low derivative gain setting. Default value is very conservative value for most systems, and is a good starting points.

Ixx32(Kvff) - Velocity FF (feed forward) Gain - Velocity feed forward gain will help the system with steady state error reduction. But whereas PID is feedback gain (causal), velocity feed forward gain is feed forward gain (non-causal). Setting it to an unreasonable value will destabilize the system.

Using the derivative gain is recommended. Often the optimal result is obtained by setting this value equal to the derivative gain value.

Ixx33(Ki) - Integral Gain - Integral gain acts to correct the system according to the accumulated following error of the system. It is particularly effective to counter the steady state error caused by friction. However, for numerical reasons (too high integral gain will “saturate” servo loop numerically) and servo stability concerns, an excessively high value of the integral gain is discouraged. Begin tuning with a lower value and observe the improvements. If you want no integral action, set it to 0.

Ixx34(IM) - Integral Mode - Setting this value to 0 enables the integrator all the time, and setting it to 1 enables the integrator only when command velocity is zero.

Ixx35(Kaff) - Acceleration FF (feed forward) Gain - This gain helps the tracking effort of the system. Determination of this gain involves somewhat complex calculations, but there are intuitive ways to apply this gain. If your velocity plot (try parabolic move for checking velocity following) of the system shows bad tracking at initial acceleration or deceleration, applying this gain will help.

Ixx29 - DAC Offset - If for some reason your amplifier or PMAC has DAC offset, it is recommended for the first step that you manually zero the DAC offset in hardware; enter 'o0' command in the terminal window and then adjust the voltage of the amplifier to zero volts by manipulating the potentiometer on the amplifier. For the second step, whatever offset that cannot be corrected by hardware adjustment may be corrected in software level by setting the DAC offset.

Ixx69 - DAC Limit - If your amplifier has more power than the motor can handle, make sure that the DAC limit is set so that excessive current will not burn the motor.

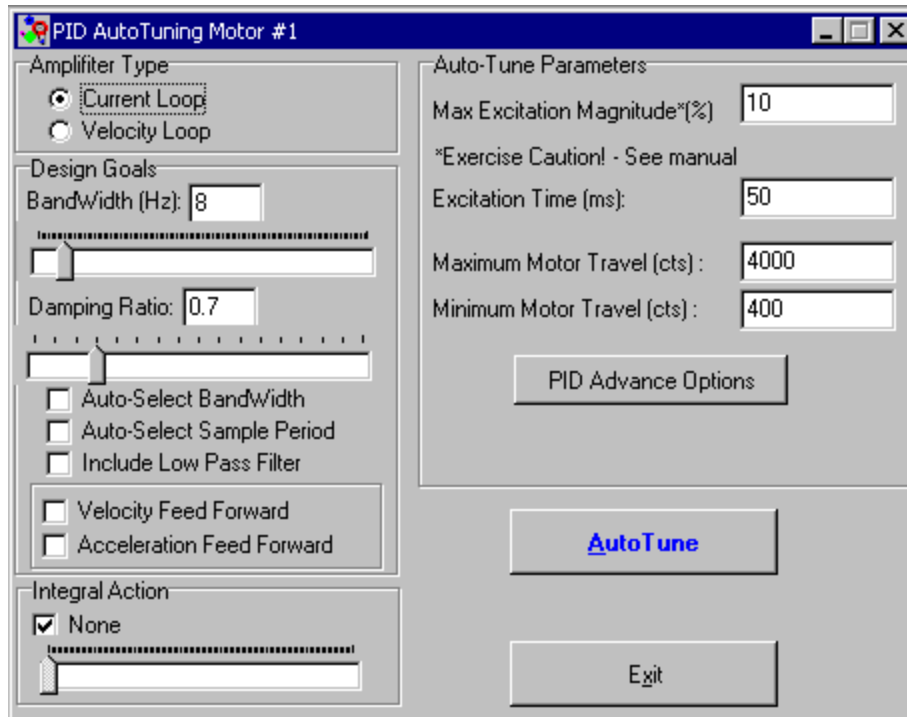
Ixx60 - Servo Cycle Extension - If the your load is moving very slowly, that is, if the encoder counts relative to the servo cycle is low all the time, it may be a good idea to extend the servo cycle. By increasing the servo cycle extension, PMAC will try to close the loops less frequently. This is desirable, since PMAC will use less time for servo calculation. More calculation power would be reserved for PLCs and other "house keeping" chores. The default value is 0, and this will update the servo loop every servo interrupt cycle. If this value is set to 1, the servo loop will be updated every 2 servo cycles, If set to 4, the servo loop will be updated every 5 servo cycles.

Ixx68 - Friction FF (feed forward) Gain - This gain acts when the position of the servo is not within 'in position' limits at zero velocity state (steady state), according to the direction needed to be compensated. Use moderation in setting this gain.

Ixx11 - Fatal Following Error Limit - When the magnitude of the following error exceeds Ix11, motor x is disabled (killed).

Regular PID Auto Tuning





Before entering the Regular PID auto-tuning session for a particular motor make sure that the motor is under closed loop control. Typically, the default gains with which the PMAC boards are shipped are low enough and may be used as the initial gains for the start of a session. Furthermore, large offset (bias) signals and deadband characteristics between the amplifier and the PMAC's DAC outputs should be reduced to a minimum by the use of I_{x29} and I_{x68} variables (see also Offset & Deadband Calibration above). When you enter the Auto-Tuning dialog box, you must select the appropriate options prior to the start of an auto-tuning session. These include:

- **Type of amplifier used with the selected motor.** (current mode or velocity mode).
- **Maximum excitation magnitude as % of DAC output.** 100% represents a DAC value of 32767 increments or 10 volts nominal. Note that depending on the type and the gain of the motor amplifier, any given excitation as % of maximum DAC output is equivalent to a certain current or voltage command to your motor. Do not demand more current or voltage than the maximum recommended value as specified for any particular the motor amplifier by its manufacturer.
- **Excitation time in milliseconds.** This should be long enough to generate about 1/4 to 1/2 of motor revolution.
- **Number of test iterations.** 2 times is usually sufficient.
- **Design goals in terms of bandwidth and damping.** Select **Auto-select Bandwidth** if you do not know how to choose a reasonable value for the closed loop bandwidth. This gives a "safe" lower bound value for the bandwidth. You may then increase the bandwidth gradually from the bandwidth automatically estimated to increase servo stiffness and auto-tune again etc.
- **The maximum and minimum travel limits for the tests.** This should be long enough to generate 1/4 to 1/2 of motor revolution and

yet short enough to avoid uncontrolled travel beyond a safe limit for a particular setup.

- **Choice of including feedforward action.** Include velocity and acceleration feedforward if you wish to minimize tracking or following errors.
- **Choice of no integral action, soft integral action or hard integral action.** Note that hard integral action should not be selected unless low bandwidth (relative to auto-select bandwidth) is selected for closed loop system. In general select soft integral action if you need zero steady state following error and a previous tuning session *without* the integral action could not satisfy your requirement.
- **Choice of automatic selection of a "safe" bandwidth based upon the test results.** Select this option if you do not know how to choose the desired closed loop bandwidth.
- **Choice of automatic selection of the servo sampling period extension I-variable ($\text{I}\times 60$) for a particular servo design goal.** It is very useful to check this option particularly when your selected closed loop bandwidth is below 20 Hz.
- **Choice of inclusion of a first order low pass filter to attenuate high frequency noise which may otherwise cause a noticeable motor jitters.**

After completing the above selection, select the **Begin Auto-Tuning** button. This prompts the simple auto-tuner to start the testing sequence. Depending on the number of iterations selected, you will observe a number of backward and forward motions. If the Pause between iterations option is checked, you will be informed of the actual distance moved in each iteration. If the box is not checked, the auto-tuner will continue testing the system as many times as selected by the user in the number of iterations input selection.

After the completion of tests, the auto-tuner computes the relevant terms of the PID Algorithm and displays them on the screen together with the existing terms for the selected motor. You may elect to discard the new terms or accept them. If accepted they are automatically downloaded to PMAC. If you notice undesirable behavior from that particular servo channel, use the Kill button to open the loop and disable the amplifier. If the motor appears stable and jitter free, then exit the Tuning menu and go to the Trajectory menu.

In the Trajectory menu you may examine the response of the system under the new set of servo controller parameters selected by the auto-tuner. Many types of moves including sinusoidal and sine sweep are available. Note that without feedforward and/or integral action there may be a considerable amount of following errors during faster moves, particularly if the selected bandwidth is relatively low.

Choices & Options in Auto-Tuning:

Amplifier Type- This selects the type of amplifier used. The default value is for a current loop amplifier. If either an analog tachometer loop is closed within the amplifier, or the amplifier operates without current feedback (in voltage mode), the Amplifier Type should be changed to velocity loop.

CAUTION

Exceeding the safe limit for DAC output specified in Maximum Excitation Magnitude for your particular setup could damage the power amplifier, the motor and the mechanical structure connected to the motor.

Maximum Excitation Magnitude- This is the size of the largest DAC signal given as a percentage value of +/- 10 volts. For example, if only two iterations are chosen for the Number of Iterations, and the Maximum Excitation Magnitude is 100%, then in the first iteration the maximum DAC output will be +/- 5 volts, and in the second iteration it will be +/- 10 volts. You must choose this parameter carefully. In general, it should be

large enough to overcome friction and other disturbances so that some motion actually does occur. On the other hand, it should not be so large as to cause amplifier, motor or mechanical damage as a result of excessive torque command. Consult your amplifier and motor specification. The default value is 30%.

Excitation Time- The period of time in milliseconds in which the DAC output is driven by the test signal is entered here. This period should be chosen carefully. It should be long enough for noticeable motion to take place (at least equal to Minimum Motor Travel). On the other hand, too long a period may result in excessive travel and/or excessive motor velocity. In most applications, excitation times between 50 and 100 milliseconds should be sufficient. For motors having very large inertia loads, large excitation times may be appropriate. The default value is 50 ms.

Number of Iterations- This refers to the number of backward/forward motions in the testing phase of an auto-tuning session. If a number greater than one is chosen, then the peak DAC output for the first iteration is determined by dividing the Maximum Excitation Magnitude by the Number of Iterations. In this way, the last iteration always corresponds to the Maximum Excitation Magnitude input. The default value is 2 iterations, which is sufficient for most applications.

WARNING

Depending on the size of the excitation signal and the responsiveness of the servo loop implemented while tuning, it is possible that the actual motor travel may go past the Maximum Motor Travel limit.

Maximum Motor Travel- You may specify independently the maximum travel limit during the tests. If this distance is reached, the loop is immediately closed and the motor is commanded to jog back towards its original position. The test will continue, however, until the specified number of iterations is completed, subject to the specified travel limit. The default value is 4000 feedback counts.

Minimum Motor Travel- This entry allows checking if in fact a detectable motion took place. If the Max Excitation magnitude is relatively low and the Number of Iterations is relatively high, the first iteration's peak DAC output may become too small. As a result no detectable motion may take place during the excitation time. By selecting an acceptable number of counts for the Minimum Motor Travel, you can safeguard against erroneous gain estimations. In general you want this entry to correspond to at least 30 degrees of motor rotation. The default value is 400 feedback counts.

Bandwidth- This entry allows specifying the closed loop system's speed of response and servo stiffness. In general, the higher the value of the bandwidth, the higher those computed terms of the ESA which correspond to the error proportional gain. In practice, typical closed loop bandwidths range from 4 Hz, for very large motor/load inertia systems, to 100 Hz for very high gain amplifiers driving extremely light motor/load inertia systems. Note that if the Auto-Select Bandwidth option is checked, the program automatically selects a lower bound "safe" bandwidth for you which will appear in this box after the tests. The default value is 20 Hz.

Damping Ratio- This entry specifies the desired closed loop system's rate decay of transient oscillations. In general, low damping ratios (ratios well below one) lead to large overshoots and long periods of transient oscillations leading towards instability. On the other hand, high damping ratios (ratios well above one) lead to very sluggish motion. Typical damping ratios should range from 0.6 to 1. Note that the higher the value of the damping ratio the higher the computed derivative action from the ESA controller, and consequently the higher the quantization noise amplification leading to possible motor jitters. In most applications a damping ratio of 0.7 to 0.8 provides a good trade off between transient response (requiring more derivative action) and quantization noise reduction (requiring less derivative action). If integral action is included, even if the specified damping ratio is unity or above, some overshoot may occur in response to a step input. Also, if the Auto-select bandwidth option is chosen, the damping ratio is set to unity. The default value of this entry is 1.

Auto-Select Bandwidth- This option is extremely useful for situations in which you do not know the ranges of acceptable values of bandwidth. When this box is checked, based upon the results of the auto-tune tests, a "safe" bandwidth is determined. This in turn selects the ESA terms and their numerical values. Typically, the "safe" bandwidth is lower than the maximum achievable bandwidth, but is more likely to lead to a jitter-free stable feedback loop.

Auto-Select Sample Period- This option, if checked, automatically selects the appropriate value of Ix60 (the "servo period extension" I-variable) suitable for the chosen bandwidth. It is advisable to check this option if a low bandwidth (less than 20 Hz) is recommended by the Auto-select Bandwidth option. By selecting this option, the servo period is extended appropriately to minimize the adverse effects of quantization noise.

Include Low Pass Filter- This option augments a first order low-pass filter to the control law which helps in the attenuation of quantization noise particularly when higher bandwidths are required and the open loop system is relatively low gain (low encoder resolution, large inertia, low amplifier gain etc.).

Velocity FeedForward- Check this box if velocity feedforward action is required. The auto-tuner will automatically determine Ix32 that would generate this action. In general, velocity feedforward is needed to overcome following errors proportional to motor velocity resulting from back emf, viscous damping, and velocity feedback. If such lags are acceptable then there will be no need for the use of this particular action and the check box may be left empty.

Acceleration FeedForward- Check this box if acceleration feedforward action is required. The auto-tuner will automatically determine Ix35 filter that would generate this action. In general, acceleration feedforward is needed to overcome following errors proportional to motor acceleration caused by inertial lags. If such lags are acceptable, then there will be no need for the use of this particular action and the check box may be left empty.

Integral Action- In general, integral action is required only if the existing amount of friction and/or torque (force) disturbances cause an unacceptable level of steady state following errors. If this happens to be the case for a particular application, then try a small amount of Integral Action (value of Ix33) and gradually increase the amount of Integral Action for faster rise time.

Pause Between Iterations- Check this box if you are interested in seeing the distance traveled during each iteration step. Usually the first time in an auto-tuning session this box is activated. Once the optimum values for the Excitation Time and the Minimum/Maximum Travel Limits are set, there will be no need to pause between test cycles or "iterations."

Doing an Auto-Tune Session:

The following provides some guidelines for typical auto-tuning activities when using the Regular PID auto-tune method.

1. Use the Jog command or otherwise move the motor to a safe position where both forward and reverse motions of at least twice the size of the **Maximum Travel Limit** are acceptable.
2. If necessary, when in the **Tuning** menu enter the **DAC Offset/Deadband Calibrate** option to adjust Ix29 and Ix68 (note that for PMAC commutated motors Ix29 should not be implemented)
3. Next select the **Auto-Tuning (simple)** option and enter its dialog box.
4. Select the correct amplifier type (*current* or *velocity loop*).
5. Enter the value of the desired bandwidth. If you are not sure how to select the value for this entry check the **Auto-Select Bandwidth** option. Select a **Damping Ratio** between 0.8 and 1.0.
6. Check **Auto-Select Sample Period** if **Auto-Select Bandwidth** is chosen and/or if the desired bandwidth is less than 20 Hz (it is assumed the value of Ix60=0 prior to entry to this dialog box).
7. For now leave the **Include Low Pass Filter** and both **Feedforward** Boxes unchecked. Also select **None** for **Integral Action**.
8. Adjust the values of **Max Excitation Magnitude**, **Excitation Time** and the **Max./Min. motor travel** so that the motor turns between 1/4 to 1/2 revolutions while accelerating during the open loop forward/reverse motions (a prolonged period of constant velocity motion should be avoided).
9. Select **Begin Auto-Tuning** button and wait for the tuner to give you the suggested values for some particular terms within the ESA. If the numbers look reasonable accept them. If the servo loop appears to behave in an unsatisfactory manner then click on the **Kill** button immediately (this action issues to the addressed motor a PMAC "kill" command). Otherwise exit this dialog box.

10. If **Auto-Select Bandwidth** was checked, enter the simple auto-tuning dialog box again, and check and record the value of bandwidth selected by the tuner. This will appear in units of Hz in the **Bandwidth** box. Then exit this dialog box.
11. Now go to the **Trajectory** menu and select **Step**. Enter the **Setup** dialog box and choose a reasonable step size (typically between 1/4 to 1/2 of motor (not necessarily load) revolution. Execute the step move.
12. Now go to the **Plotting** menu and select *Actual position* and *commanded position* both on the left hand side and plot. If the natural frequency evaluated from the step response is within 10 to 20 % of the selected bandwidth the tuner has done a good job.
13. If the reported natural frequency is not close enough (say within 10 to 20%) to the desired bandwidth selected in the auto-tune dialog box, try tuning with more/less excitation magnitudes and excitation times.
14. If the step response is satisfactory, go back to the **Auto-Tune (simple)** dialog box and if there is a constant following error add some **Integral** action. Check the **feedforward** actions if tracking errors are to be minimized, and **Low Pass Filter** if noise induced jitters are noticeable. Re-tune and implement the new controller, then exit the dialog box.
15. Now go back to the **Trajectory** menu and, this time, choose a trajectory which is typical of your application moves. Execute it and plot the response. If the response is not satisfactory due to unacceptable following errors, re-enter the tuner and increase the desired bandwidth. Alternatively if the response is too noisy, reduce the desired bandwidth and auto-tune again until a satisfactory response is obtained. A good trajectory to try first is the *Parabolic Velocity*.
16. If there still exists some noticeable "ringing" at a fairly constant period which appears to decay with time and has a frequency not more than 10 times the closed loop bandwidth, you may try the **Notch Filter** option to implement a *notch filter*. On the other hand, if there are persistent high frequency jitters (buzzing), you should try to reduce the **Bandwidth** and/or the **Damping Ratio** and re-tune again in order to decrease the servo response to this "noise."

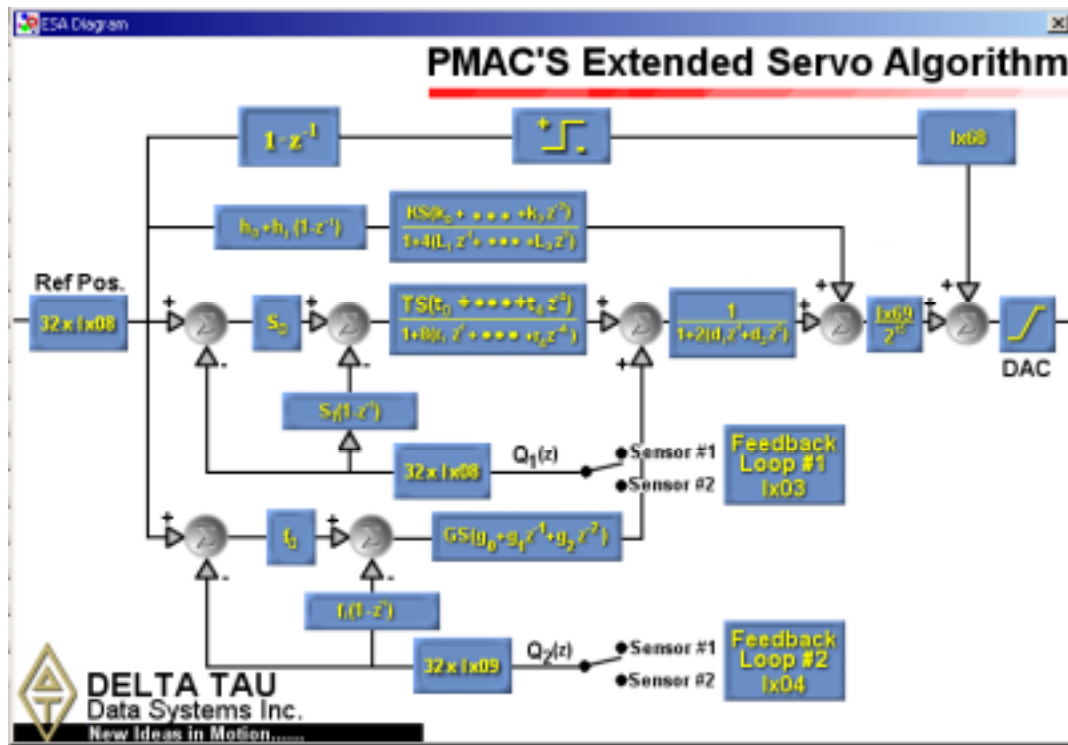
CAUTION

Depending on the controller gain, for even small amplitudes of the sine wave, the DAC output under closed loop control may oscillate at its maximum magnitude during the sine sweep, particularly at higher frequencies. This may cause motor, amplifier and/or mechanical structure damage. Do not execute a sine sweep trajectory with a high max frequency if you are uncertain about the ability of your electromechanical hardware to withstand the current or torque demands generated by the sine sweep at high frequencies.

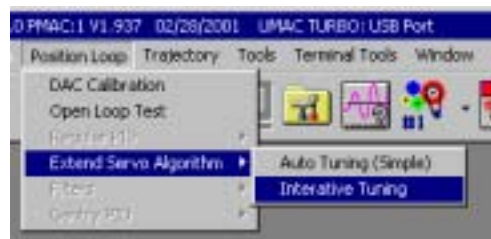
17. Finally, you may elect to obtain the *frequency response* of the closed loop system, by selecting the **Sine Sweep** trajectory under the **Trajectory** menu. Select a maximum frequency not more than 5 times the close loop bandwidth with a magnitude not greater than 1/4 of revolution of your motor with a period not more than 10 seconds. After the end of the sweep go to the **Plotting** menu, select **Code Plot** and then plot the frequency response for both *magnitude* and *phase*.

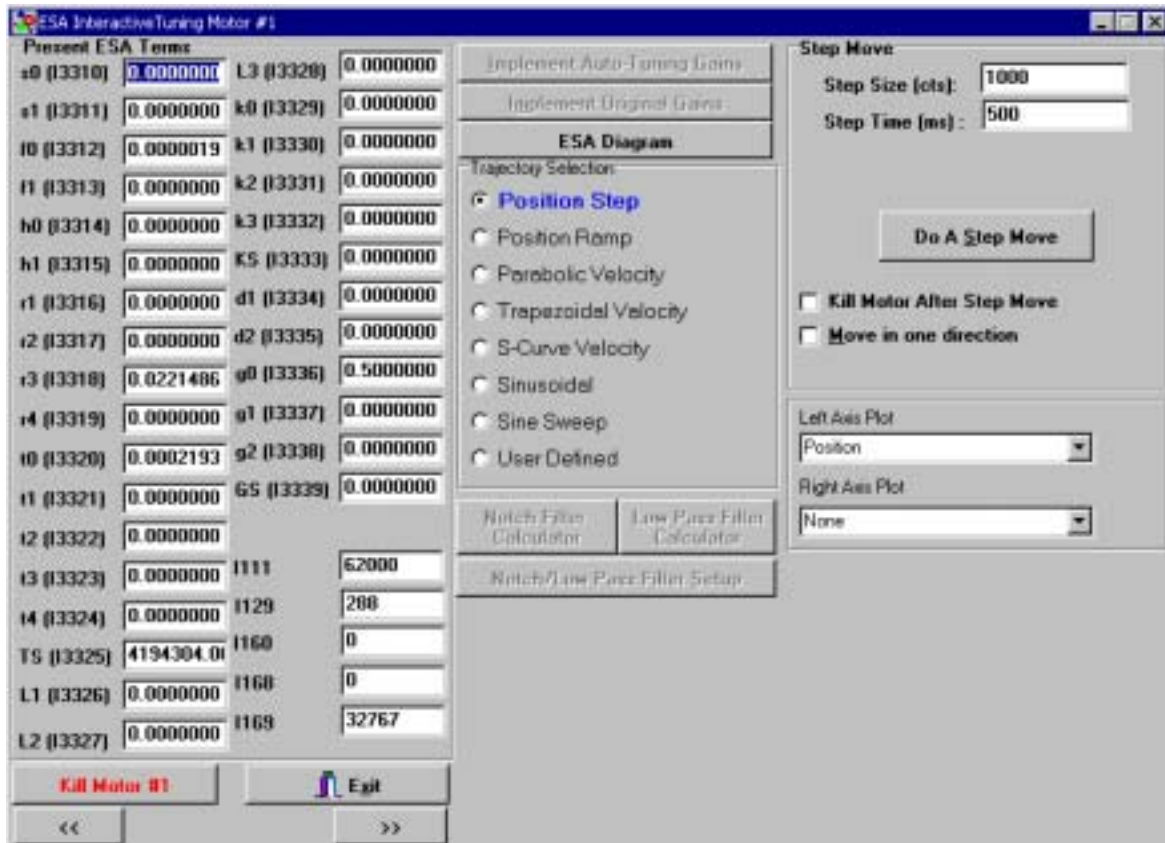
Once a satisfactory set of servo parameters is obtained, you may elect to save them on PMAC with the on-line "SAVE" command via the PMAC Terminal.

Extended Servo Algorithm



ESA Interactive Tuning





ESA Structure

Whenever a PMAC is ordered with its option 6 or a Turbo PMAC's Iyy00 is set to 1, the standard PID servo algorithm for each of the servo channels is replaced by ESA. There are two feedback loops and one feedforward loop. Typically, in a dual feedback application, the data from the sensor connected to the load would be directed (via PMAC's "Encoder Conversion Table") to the feedback loop #1. The data from the feedback sensor connected to the actuator would be brought into the loop through loop #2. There are two feedforward paths. The dynamic feedforward path operates directly on the reference (commanded) motor position which includes the "hand wheel" input. This path can be used to compensate for dynamic lags which may otherwise cause following (tracking) errors. The second feedforward path contains a simple "sign of commanded velocity" nonlinear operator which is intended to be used for dry (static and/or sliding) friction compensation.

Structure of ESA Diagram

The following table specifies the corresponding I-variable assignment and the numerical range of the discrete-time polynomials within ESA.

I-variable	Symbol	Numeric Type	Numerical Range*
Ix30	s ₀	real	-1.0 to 0.99999988
Ix31	s ₁	real	-1.0 to 0.99999988
Ix32	f ₀	real	-1.0 to 0.99999988
Ix33	f ₁	real	-1.0 to 0.99999988
Ix34	h ₀	real	-1.0 to 0.99999988
Ix35	h ₁	real	-1.0 to 0.99999988
Ix36	r ₁	real	-1.0 to 0.99999988
Ix37	r ₂	real	-1.0 to 0.99999988
Ix38	r ₃	real	-1.0 to 0.99999988
Ix39	r ₄	real	-1.0 to 0.99999988
Ix40	t ₀	real	-1.0 to 0.99999988
Ix41	t ₁	real	-1.0 to 0.99999988
Ix42	t ₂	real	-1.0 to 0.99999988
Ix43	t ₃	real	-1.0 to 0.99999988
Ix44	t ₄	real	-1.0 to 0.99999988
Ix45	TS	integer	-2 ²³ to 2 ²³ -1
Ix46	L ₁	real	-1.0 to 0.99999988
Ix47	L ₂	real	-1.0 to 0.99999988
Ix48	L ₃	real	-1.0 to 0.99999988
Ix49	k ₀	real	-1.0 to 0.99999988
Ix50	k ₁	real	-1.0 to 0.99999988
Ix51	k ₂	real	-1.0 to 0.99999988
Ix52	k ₃	real	-1.0 to 0.99999988
Ix53	KS	integer	-2 ²³ to 2 ²³ -1
Ix54	d ₁	real	-1.0 to 0.99999988
Ix55	d ₂	real	-1.0 to 0.99999988
Ix56	g ₀	real	-1.0 to 0.99999988
Ix57	g ₁	real	-1.0 to 0.99999988
Ix58	GS	integer	-2 ²³ to 2 ²³ -1
Ix68	no symbol	integer	0 to 2 ¹⁶ -1
* The internal word length used for all of the above I-variables is 24 bits.			

I-variable Assignment & Numerical Specification of PMAC's ESA Controller

Other scaling factors include the usual 32*Ix08 multiplied by the commanded position and loop #1 actual position, 32*Ix09 multiplied by the Loop #2 actual position, 32*Ix07 multiplied by the "master hand wheel" input. In addition, the DAC scale factor is Ix69/2¹⁵. Note that the control effort is limited to 16-bits prior to final Ix69/2¹⁵ scaling. As a result, Ix69 is now essentially a scale factor which also limits

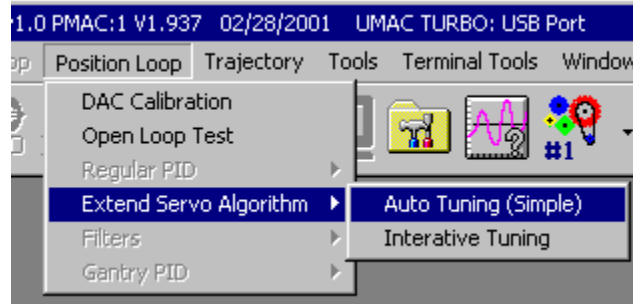
the control effort. Therefore, the function of I_{x69} is somewhat different in relation to the standard PMAC PID servo loop. For example if I_{x69} is set equal to 16384, then the maximum magnitude of the DAC value is limited to 16384 (+/- 5 volts), and the numerical value of the control effort is multiplied by 0.5 prior to updating the DAC output (or the commutation algorithm input whenever it is active).

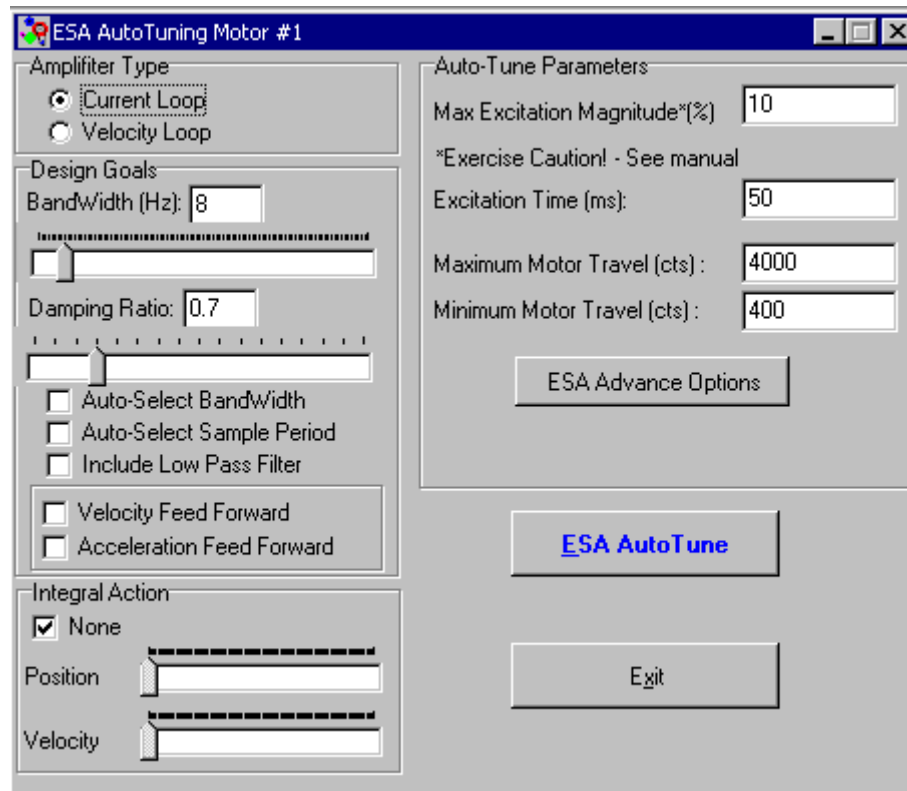
Also, the servo cycle extension I-variable, I_{x60} , is no longer restricted to numerical values defined by $2^n - 1$. In fact I_{x60} can be any integer number. This feature allows evenly adjustable servo sampling selection in sub-multiples of the base servo interrupt frequency. For example, when $I_{160}=0$, $I_{260}=1$ and $I_{360}=4$, then the servo loop for channel 1 runs at the servo interrupt frequency, the servo loop for channel 2 is executed at half the servo interrupt frequency, and the servo loop for channel 3 runs at 1/5 of the servo interrupt frequency. The servo interrupt frequency is itself hardware jumper selected (refer to PMAC's hardware reference manual).

The Deadband/dry friction compensation I-variable (I_{x68}) may be set to a positive integer number in the range of 0 to 65535. At every servo cycle the magnitude of I_{x68} is simply added or subtracted from the instantaneous control effort, depending on the sign of the commanded velocity.

Finally, other auxiliary servo parameter I-variables of the standard PMAC PID in the range of I_{x63} to I_{x67} are not implemented in the ESA version.

ESA Auto Tuning





The Auto-Tuning option brings up the dialog box shown above. The inputs, check boxes and options within this dialog box are similar to the Auto-Tuning dialog box of the Regular PID algorithm.

Prior to entering the ESA auto-tuning session for a particular motor, make sure that the motor is under closed loop control. The default gains with which the PMAC boards are shipped typically are low enough and may be used as the initial gains for the start of a session. Furthermore, large offset (bias) signals and deadband characteristics between the amplifier and the PMAC's DAC outputs should be reduced to a minimum by the use of Ix29 and Ix68 variables. When you enter the ESA Auto-Tuning dialog box, you must select the appropriate options prior to the start of an auto-tuning session. These include:

- **Type of amplifier used with the selected motor.** (current mode or velocity mode)
- **Maximum excitation magnitude as % of DAC output.** 100% represents a DAC value of 32767 increments or 10 volts nominal. Note that depending on the type and the gain of the motor amplifier, any given excitation as % of maximum DAC output is equivalent to a certain current or voltage command to your motor. One should not demand more current or voltage than the maximum recommended value as specified for any particular the motor amplifier by its manufacturer.
- **Excitation time in milliseconds.** Should be long enough to generate about 1/4 to 1/2 of motor revolution.
- **Number of test iterations.** 2 times is usually sufficient.
- **Design goals in terms of bandwidth and damping.** Select **Auto-select Bandwidth** if you do not know how to choose a reasonable value for the closed loop bandwidth. This

gives a "safe" lower bound value for the bandwidth. You may then increase the bandwidth gradually from the bandwidth automatically estimated to increase servo stiffness and auto-tune again etc.

- **The maximum and minimum travel limits for the tests.** This should be long enough to generate 1/4 to 1/2 of motor revolution and yet short enough to avoid uncontrolled travel beyond a safe limit for a particular setup.
- **Choice of including feedforward action.** Include velocity and acceleration feedforward if you wish to minimize tracking or following errors.
- **Choice of no integral action, soft integral action or hard integral action.** Note that hard integral action should not be selected unless low bandwidth (relative to auto-select bandwidth) is selected for closed loop system. In general select soft integral action if you need zero steady state following error and a previous tuning session *without* the integral action could not satisfy your requirement.
- **Choice of automatic selection of a "safe" bandwidth based upon the test results.** Select this option if you do not know how to choose the desired closed loop bandwidth.
- **Choice of automatic selection of the servo sampling period extension I-variable (1x60) for a particular servo design goals.** It is useful to check this option, particularly when your selected closed loop bandwidth is below 20 Hz.
- **Choice of inclusion of a first order low pass filter to attenuate high frequency noise which may otherwise cause a noticeable motor jitters.**

After completing the above selection, select the **Begin Auto-Tuning** button. This prompts the simple auto-tuner to start the testing sequence. Depending on the *number of iterations* selected, you will observe a number of backward and forward motions. If the **Pause between iterations** option is checked, you will be informed of the actual distance moved in each iteration. If the box is not checked the auto-tuner will continue testing the system as many times as selected by the user in the *number of iterations* input selection.

After the completion of tests, the auto-tuner computes the relevant terms of the ESA and displays them on the screen together with the existing terms for the selected motor. You may elect to discard the new terms or accept them. If accepted they are automatically downloaded to PMAC. If you notice undesirable behavior from that particular servo channel use the **Kill** button to open the loop and disable the amplifier.

In the **Trajectory** menu you may examine the response of the system under the new set of servo controller parameters selected by the auto-tuner. Many types of moves including *sinusoidal* and *sine sweep* are available. Note that without feedforward and/or integral action there may be a considerable amount of following errors during faster moves, particularly if the selected bandwidth is relatively low.

The new ESA parameters are not permanently saved within PMAC, and will be lost upon reset or power down. When you have reached a satisfactory set of gains you may save them within PMAC by opening up the **PMAC Terminal** under the **Utility** menu and typing PMAC on line command "SAVE."

Choices & Options in ESA Auto-Tuning:

Amplifier Type- This selects the type of amplifier used. The default value is for a *current loop* amplifier. If either an analog tachometer loop is closed within the amplifier or, the amplifier operates without current feedback (in voltage mode), the **Amplifier Type** should be changed to *velocity loop*.

CAUTION

Do not exceed the safe limit for DAC output specified in Maximum Excitation Magnitude for your particular setup. Otherwise you may damage the power amplifier, the motor and the mechanical structure connected to the motor.

Maximum Excitation Magnitude- This is the size of the largest DAC signal, given as a percentage value of +/- 10 volts. For example if only two iterations are chosen for the **Number of Iterations**, and the **Maximum Excitation Magnitude** is 100%, then in the first iteration the maximum DAC output will be +/- 5 volts, in the second iteration it will be +/- 10 volts. You must choose this parameter carefully. In general, it should be large enough to overcome friction and other disturbances so that actually some motion does occur. On the other hand, it should not be too large to cause amplifier, motor or, mechanical damage as a result of excessive torque command. Consult your amplifier and motor specification. The default value is 30%.

Excitation Time- The period of time (in milliseconds) in which the DAC output is driven by the test signal is entered here. Choose this period carefully. It should be long enough for noticeable motion to take place (at least equal to **Minimum Motor Travel**). On the other hand too long a period of time may mean excessive travel and/or excessive motor velocity. In most applications, excitation times between 50 to 100 milliseconds should be sufficient. For motors having very large inertia loads, large excitation times may be appropriate. The default value is 50 ms.

Number of Iterations- This refers to the number of backward/forward motions in the testing phase of an auto-tuning session. If a number greater than one is chosen, then the peak DAC output for the first iteration is determined from dividing the **Maximum Excitation Magnitude** by the **Number of Iterations**. In this way, the last iteration always corresponds to the Maximum Excitation Magnitude input. The default value is 2 iterations, which is sufficient for most applications.

WARNING

Depending on the size of the excitation signal and the responsiveness of the servo loop implemented while tuning, it is possible that the actual motor travel may go past the Maximum Motor Travel limit.

Maximum Motor Travel- You may specify independently the maximum travel limit during the tests. If this distance is reached, the loop is immediately closed and the motor is commanded to jog back towards its original position. The test will continue, however, until the specified number of iterations is completed subject to the specified travel limit. The default value is 4000 feedback counts.

Minimum Motor Travel- This entry allows checking if in fact a detectable motion took place. If the **Max Excitation magnitude** is relatively low and the **Number of Iterations** is relatively high, the first iteration's peak DAC output may become too small. As a result, no detectable motion may take place during the excitation time. By selecting an acceptable number of counts for the **Minimum Motor Travel**, you can safeguard against erroneous gain estimations. In general you want this entry to correspond to at least 30 degrees of motor rotation. The default value is 400 feedback counts.

Bandwidth- This entry allows specifying the closed loop system's speed of response and servo stiffness. In general, the higher the value of the bandwidth, the higher those computed terms of the ESA which correspond to the error proportional gain. In practice, typical closed loop bandwidths range from 4 Hz, for very large motor/load inertia systems, to 100 Hz for very high gain amplifiers driving extremely light motor/load inertia systems. Note that if the **Auto-Select Bandwidth** option is checked, the program automatically select a lower bound "safe" bandwidth for you which will appear in this box after the tests. The default value is 20 Hz.

Damping Ratio- This entry specifies the desired closed loop system's decay rate of transient oscillations. In general, low damping ratios (ratios well below one) lead to large overshoots and long periods of transient

oscillations, leading towards instability. On the other hand, high damping ratios (ratios well above one) lead to very sluggish motion. Typical damping ratios should range between 0.6 to 1. Note that the higher the value of the damping ratio, the higher the computed derivative action from the ESA controller and consequently the higher the quantization noise amplification leading to possible motor jitters. In most applications a damping ratio of 0.7 to 0.8 provides a good trade-off between transient response (requiring more derivative action) and quantization noise reduction (requiring less derivative action). If integral action is included, even if the specified damping ratio is unity or above, some overshoot may occur in response to a step input. Also, if the **Auto-select bandwidth** option is chosen, the damping ratio is set to unity. The default value of this entry is 1.

Auto-select Bandwidth- This option is useful for situations in which you do not know the ranges of acceptable values of bandwidth. When, this box is checked, based upon the results of the auto-tune tests, a "safe" bandwidth is determined. This in turn selects the ESA terms and their numerical values. The "safe" bandwidth typically is lower than the maximum achievable bandwidth, but is more likely to lead to a jitter free stable feedback loop.

Auto-select Sample Period- This option, if checked, automatically selects the appropriate value of I_{x60} (the "servo period extension" I-variable) suitable for the chosen bandwidth. It is advisable to check this option if a low bandwidth (less than 20 Hz) is recommended by the **Auto-select Bandwidth** option. When this option is selected, the servo period is extended appropriately to minimize the adverse effects of quantization noise.

Include Low Pass Filter- This option augments a first order low pass filter to the control law which helps in the attenuation of quantization noise particularly when higher bandwidths are required and the open loop system is relatively low gain (low encoder resolution, large inertia, low amplifier gain etc.).

Velocity FeedForward- Check this box if velocity feedforward action is required. The auto-tuner will determine automatically the terms in the ESA feedforward filter that would generate this action. In general, velocity feedforward is needed to overcome following errors proportional to motor velocity resulting from back emf, viscous damping, and velocity feedback. If such lags are acceptable then there will be no need for the use of this particular action and the check box may be left empty.

Acceleration FeedForward- Check this box if acceleration feedforward action is required. The auto-tuner will automatically determine the terms in the ESA feedforward filter that would generate this action. In general, acceleration feedforward is needed to overcome following errors proportional to motor acceleration caused by inertial lags. If such lags are acceptable, then there will be no need for the use of this particular action and the check box may be left empty.

Integral Action- In general, integral action is only required if the existing amount of friction and/or torques (force) disturbances cause an unacceptable level of steady state following errors. If this happens to be the case for a particular application, then try first the *soft* integral action. Use the *hard* integral action only if the selected bandwidth is low relative to the value recommended by the **Auto-select Bandwidth** option.

Pause Between Iterations- Check this box if you are interested to see the distance traveled during each iteration step. Usually the first time in an auto-tuning session this box is activated. Once the optimum values for the **Excitation Time** and the **Minimum/Maximum Travel Limits** are set, there will be no need to pause between test cycles or "iterations."

Doing an ESA Auto-Tune Session:

The following provides some guidelines for typical auto-tuning activities when using the "simple" auto-tune method.

1. Using the `Jog` command or otherwise move the motor to a safe position where both forward and reverse motions of at least twice the size of the **Maximum Travel Limit** are acceptable.
2. If necessary, when in the **Tuning** menu enter the **DAC Offset/Deadband Calibrate** option to adjust I_{x29} and I_{x68} . For PMAC commutated motors, I_{x29} should not be implemented.
3. Select the **Auto-Tuning (simple)** option and enter its dialog box.
4. Select the correct amplifier type (*current* or *velocity loop*).
5. Enter the value of the desired bandwidth. If you are not sure how to select the value for this entry check the **Auto-Select Bandwidth** option. Select a **Damping Ratio** between 0.8 and 1.0.

6. Check **Auto-Select Sample Period** if **Auto-Select Bandwidth** is chosen and/or if the desired bandwidth is less than 20 Hz (it is assumed the value of $1 \times 60 = 0$ prior to entry to this dialog box).
7. For now leave the **Include Low Pass Filter** and both **Feedforward** Boxes unchecked. Also select **None** for **Integral Action**.
8. Adjust the values of **Max Excitation Magnitude**, **Excitation Time** and the **Max./Min. motor travel** so that the motor turns from 1/4 to 1/2 revolutions while accelerating during the open loop forward/reverse motions (a prolonged period of constant velocity motion should be avoided).
9. Select the **Begin Auto-Tuning** button and wait for the tuner to give you the suggested values for some particular terms within the ESA. If the numbers look reasonable, accept them. If the servo loop appears to behave in an unsatisfactory manner then click on the **Kill** button immediately. This action issues to the addressed motor a PMAC "kill" command). Otherwise exit this dialog box.
10. If **Auto-Select Bandwidth** was checked, enter the simple auto-tuning dialog box again, and check and record the value of bandwidth selected by the tuner (this will appear in units of Hz in the **Bandwidth** box). Then exit this dialog box.
11. Now go to the **Trajectory** menu and select **Step**. Enter the **Setup** dialog box and choose a reasonable step size (typically from 1/4 to 1/2 of motor (not necessarily load) revolution. Execute the step move.
12. Now go to the **Plotting** menu and select *Outer loop position* and *commanded position*, both on the left-hand side and plot. If the natural frequency evaluated from the step response is within 10 to 20 % of the selected bandwidth the tuner has done a good job.
13. If the reported natural frequency is not close enough (say within 10 to 20%) to the desired bandwidth selected in auto-tune dialog box, try tuning with more/less excitation magnitudes and excitation times.
14. If the step response is satisfactory, go back to the **Auto-Tune (simple)** dialog box and if there is a constant following error add **Soft Integral** action. Check the **feedforward** actions if tracking errors are to be minimized, and **Low Pass Filter** if noise induced jitters are noticeable. Re-tune and implement the new controller. Then exit the dialog box.
15. Now go back to the **Trajectory** menu, and, this time choose a trajectory that is typical of your application moves. Execute it and plot the response. If the response is not satisfactory due to unacceptable following errors, re-enter the tuner and increase the desired bandwidth. Alternatively if the response is too noisy, reduce the desired bandwidth and auto-tune again until a satisfactory response is obtained. A good trajectory to try first is the *Parabolic Velocity*. The features of this trajectory are identical to the *Parabolic* option in the PID tuning dialog box of the PMAC Executive program.
16. If there still exists some noticeable "ringing" at a fairly constant period which appears to decay with time and has a frequency not more than 10 times the closed loop bandwidth, you may try the **Notch Filter** option to implement a *notch filter*. On the other hand, if there are persistent high frequency jitters (buzzing), you should try to reduce the **Bandwidth** and/or the **Damping Ratio** and re-tune again in order to decrease the servo response to this "noise".

CAUTION

Depending on the controller gain, for even small amplitude of the sine wave, the DAC output under closed loop control may oscillate at its maximum magnitude during the sine sweep particularly at higher frequencies. This may cause motor, amplifier and/or mechanical structure damage. ***Do not execute a sine sweep trajectory with a high max. frequency if you are uncertain about the ability of your electromechanical hardware to withstand the current or torque demands generated by the sine sweep at high frequencies.***

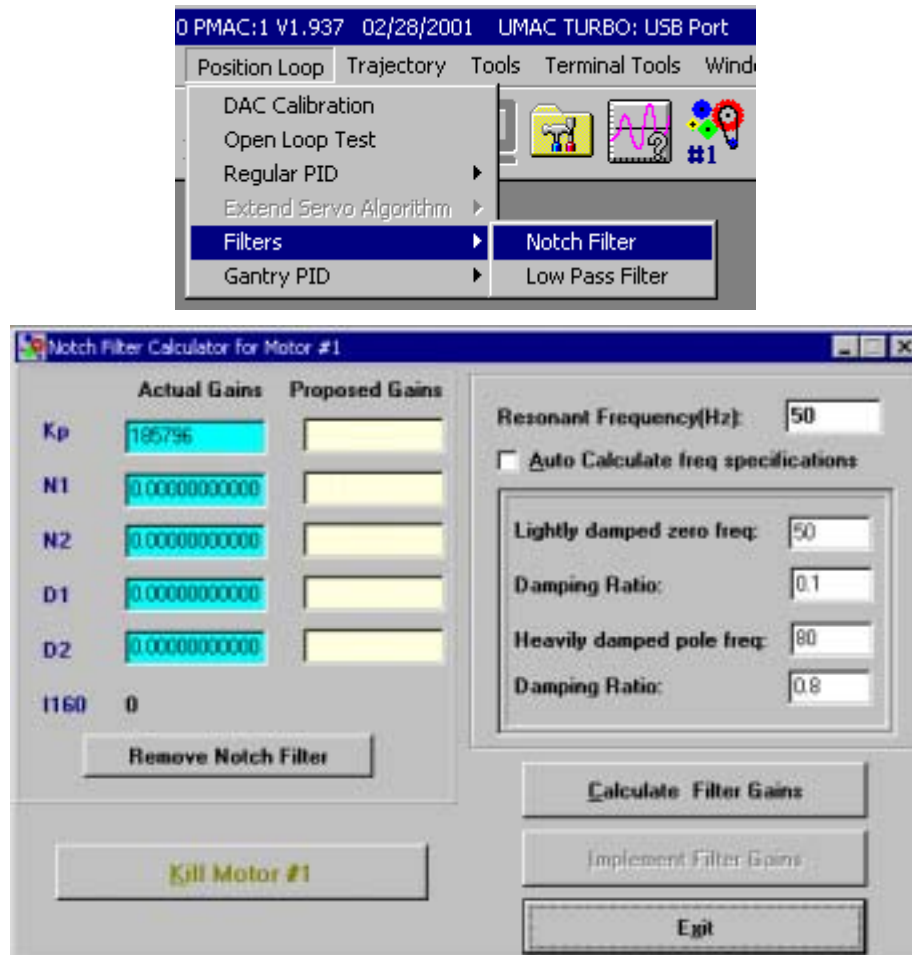
17. Finally, you may elect to obtain the *frequency response* of the closed loop system, by selecting the **Sine Sweep** trajectory under the **Trajectory** menu. Select a maximum frequency not more than 5 times the close loop bandwidth with a magnitude not greater than 1/4 of revolution of your motor

with a period not more than 10 seconds. After the end of the sweep go to the **Plotting** menu, select **Bode Plot** and then plot the frequency response for both *magnitude* and *phase*. Once satisfactory servo parameters are obtained, you may elect to save them on PMAC using the on-line "SAVE" command via the PMAC Terminal option under the Utilities menu and/or backup the I-variables on the host PC via the Back-Up PMAC I-variables option under the Utilities menu .

Filter

Notch Filter (Addition to PID +Feedforward)

PMAC Tuning allows you to set up a notch filter very easily, without the need to understand how a notch filter works.



First, select a motor using <PgUp> or <PgDn>. Next, enter the frequency of the mechanical resonance that you wish to notch out. If you have the Auto Calculate Frequency Specifications checkbox selected, then all you need to do is select IMPLEMENT NOTCH FILTER. The notch filter coefficients (Ix36 - Ix39) will be calculated automatically and displayed on the left, and these new coefficients will be downloaded to PMAC.

Resonant Frequency. This value is the actual frequency (in hertz) you wish to notch out in the servo loop. To determine which frequency this is, you may need to do a step move from the tuning screen and measure this frequency from looking at the plot of the step response.

Auto-Calculate Frequency Specifications. This checkbox allows you to let the Executive determine the necessary other frequencies (and their respective damping ratios) in order to calculate the notch filter coefficients. In most cases, you will want this option selected.

Lightly Damped Zero Frequency. In practice, this frequency is typically a value 90% of the resonant frequency. Remember that this value is calculated for you if you have the Auto Calculate Frequency Specifications checkbox selected.

Lightly Damped Zero Frequency Damping Ratio. In practice, this damping ratio is typically a value of 0.2. Remember that this value is calculated for you if you have the Auto Calculate Frequency Specifications checkbox selected.

Heavily Damped Pole Frequency. In practice, this frequency is typically a value 145% of the resonant frequency. Remember that this value is calculated for you if you have the Auto Calculate Frequency Specifications checkbox selected.

Heavily Damped Pole Frequency Damping Ratio. In practice, this damping ratio is typically a value of 0.8. Remember that this value is calculated for you if you have the Auto Calculate Frequency Specifications checkbox selected.

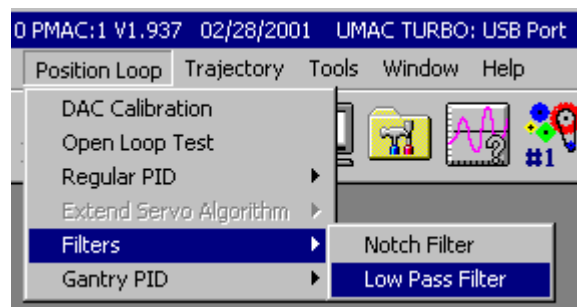
Remove Notch Filter. This button removes the notch filter for the selected motor by zeroing out the coefficients (Ix36 - Ix39) and restoring Ix30 to its near original value (before the notch filter was implemented).

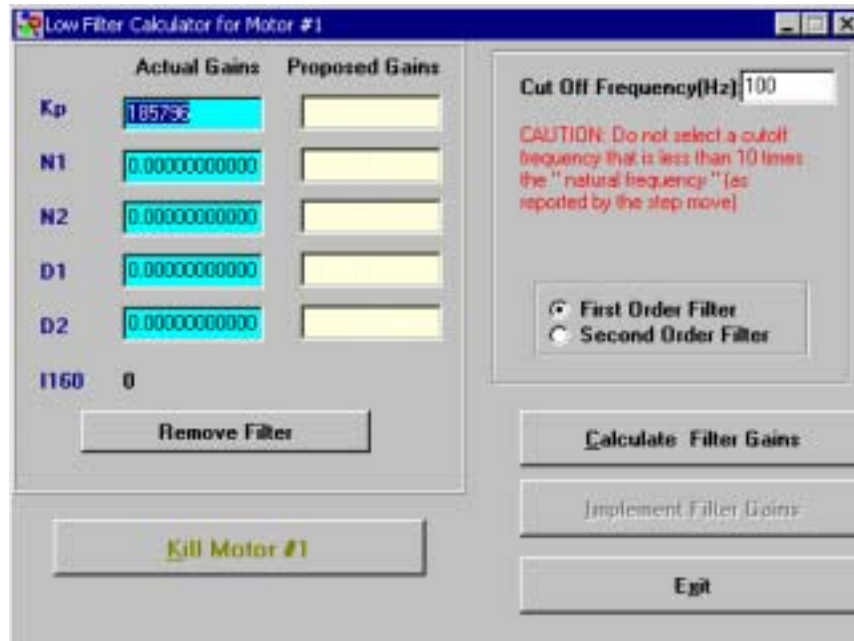
Calculate Notch Filter Gain. This button calculates the coefficients for the notch filter for the selected motor and displays the values on the left side of the screen. The new values are not downloaded to PMAC. When you calculate a notch filter, you will see the proportional gain (Ix30) change to a higher value. This is necessary to maintain the same DC gain with the new notch filter implemented. The amount Ix30 changes will depend on the values of the notch filter coefficients.

Implement Notch Filter. This button calculates the coefficients for the notch filter for the selected motor and displays the values on the left side of the screen. The new values are downloaded to PMAC. When you implement a notch filter, you will see the proportional gain (Ix30) change to a higher value. This is necessary to maintain the same DC gain with the new notch filter implemented. The amount Ix30 changes will depend on the values of the notch filter coefficients.

Low Pass Filter (Addition to PID + Feedforward)

PMAC Tuning allows you to set up a low pass filter easily, without the need to understand how a low pass filter works.





First, select a motor using <PgUp> or <PgDn>. Next, enter the cutoff frequency and select either a 1st order or 2nd order filter. Now all you need to do is select IMPLEMENT LOW PASS FILTER. The low pass filter coefficients (Ix38 and Ix39) will be calculated automatically and displayed on the left, and these new coefficients will be downloaded to PMAC.

Cutoff Frequency. This value is the actual cutoff frequency (in hertz) you wish to use for the low pass filter in the servo loop. To determine which frequency this is, you may need to do a step move from the tuning screen and measure this frequency from looking at the plot of the step response.

First Order Filter. This radio button allows you to specify a 1st order lowpass filter. 1st order filters typically introduce little lag to your servo but do not exhibit a steep cutoff for the pass band.

Second Order Filter. This radio button allows you to specify a 2nd order lowpass filter. 2nd order filters typically introduce more lag to your servo but do exhibit a good, steep cutoff for the pass band.

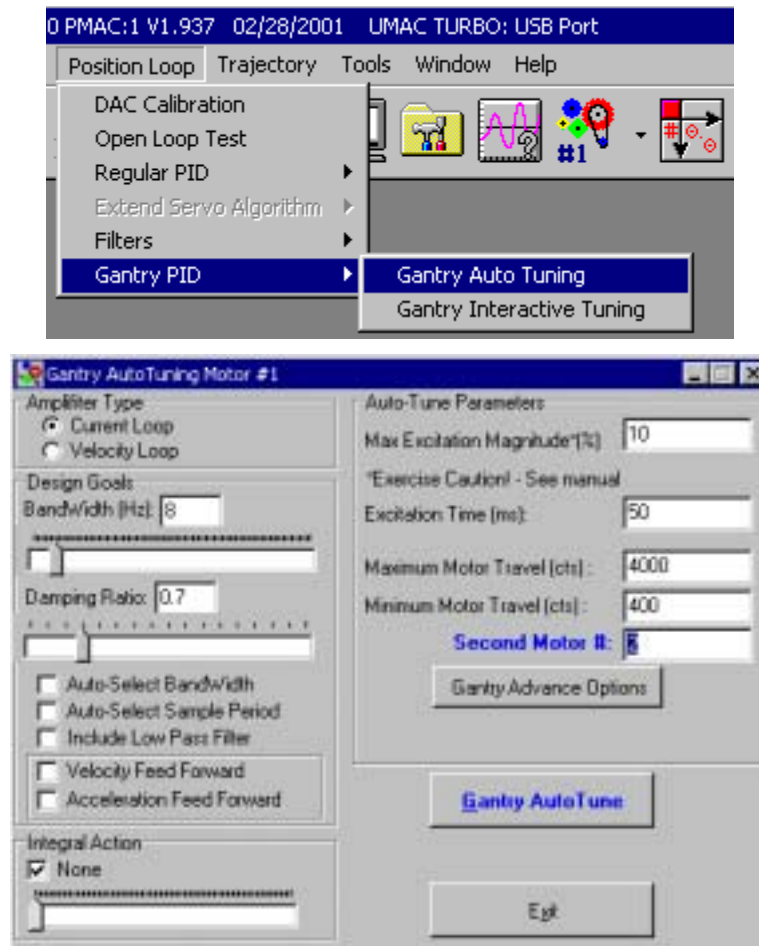
Remove Low Pass Filter. This button removes the low pass filter for the selected motor by zeroing out the coefficients (Ix38 and Ix39) and restoring Ix30 to its near original value (before the low pass filter was implemented).

Calculate Low Pass Filter. This button calculates the coefficients for the low pass filter for the selected motor and displays the values on the left side of the screen. The new values are not downloaded to PMAC. When you calculate a low pass filter, you will see the proportional gain (Ix30) change to a lower value. This is necessary to maintain the same DC gain with the new low pass filter implemented. The amount Ix30 changes will depend on the values of the low pass filter coefficients.

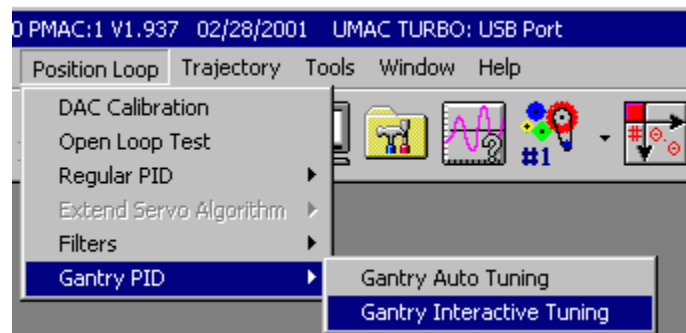
Implement Low Pass Filter. This button calculates the coefficients for the low pass filter for the selected motor and displays the values on the left side of the screen. The new values are downloaded to PMAC. When you implement a low pass filter, you will see the proportional gain (Ix30) change to a lower value. This is necessary to maintain the same DC gain with the new low pass filter implemented. The amount Ix30 changes will depend on the values of the low pass filter coefficients.

Gantry PID

Gantry Auto Tuning



Gantry Interactive Tuning





CAUTION

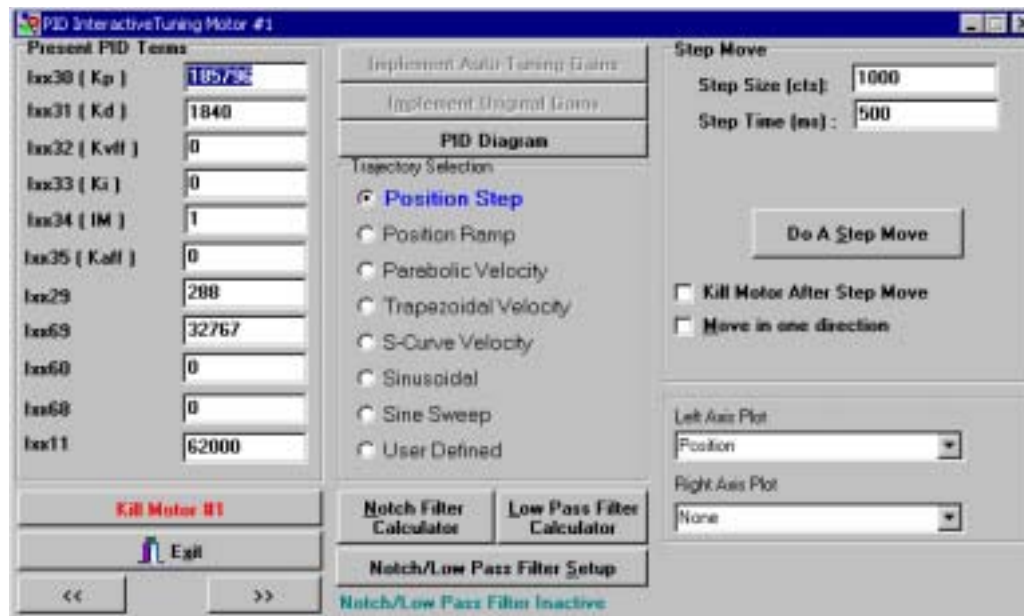
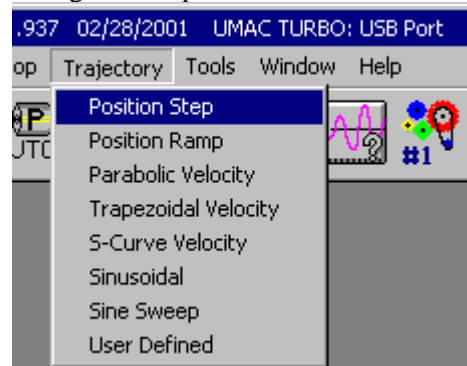
If the selected Second Motor is not well tuned (relative to the motor being tuned) the response of the Gantry system may be unsatisfactory to the extent that physical damage may occur due to mechanical interaction.

The Gantry Interactive and Auto Tuning has been implemented for Regular PID Algorithm only. For both Interactive and Auto Tuning, the dialog boxes are identical to the Regular PID Algorithm described above except that a selected Second Motor has to be specified.

For Interactive Gantry Tuning the selected Second Motor position loop will be closed (if open) and it will be commanded to move the same distance. It is recommended that prior to any commanded moves both motors be assigned with the same or similar Ix30 and Ix31 gains, and that as the tuning proceeds towards stiffer servo (higher) gains for the primary motor, the Second Motor gains are also increased so that the dynamic response of both motors remains similar.

CHAPTER 5 - TRAJECTORIES

The trajectory menu allows the executing several pre-defined as well as custom defined moves.



All of these moves and corresponding plots are explained in the following sections:

Position Step

(Feedback Tuning with Step Response)

Step response is often used as a method of evaluating a feedback filter. Many controls textbooks contain information on interpreting step responses for establishing proper feedback, particularly for second-order systems (Current-controlled motors driving inertial loads are second-order systems.) In a step response, a sudden change is made to the command position and the feedback filter attempts to bring the system to this new position. In observing how the system gets to the new position, we can deduce a great deal about the properties of the system. It does not matter that you will not ever create such a large instantaneous step in position in the actual operation of your system. The purpose of this "jolt to the system" is to bring out system characteristics that might otherwise not be obvious. This detailed information on the PID filter is not essential to performing the tuning, but is included here for reference.

PMAC has three feedback parameters to be adjusted in this process:

- Kp: Proportional gain (Ix30)
- Kd: Derivative gain (Ix31)
- Ki: Integral gain (Ix33)

We will be looking at three key step response parameters to set the feedback:

Rise Time. The time it takes for the system to go from 10% to 90% of the commanded step. Natural frequency is directly related to this.

Overshoot. The percentage past the commanded step that the system travels. Damping ratio is directly related to this.

Settling time. The time it takes for the system to get and stay within 5% of the commanded step.

Typically, what is desired is a quick rise time with little or no overshoot and quick settling time. The case of critical damping, which is the fastest possible rise time that creates no overshoot, is often the goal. There are usually tradeoffs between these parameters, particularly between fast response and low overshoot. If your amplifier has a tachometer, the tachometer is providing derivative gain (and therefore damping) within the amplifier itself. If the amplifier has been well tuned, you should not have to add any more derivative gain in the digital filter, but you are free to do so if you wish. On PMAC, it is possible to have the error integration active at all times by setting Ix34 to 0, or to have it active only when the motion is stopped by setting Ix34 to 1. While the step response for these two cases will look essentially identical, the behavior on real moves will be very different. Error integration that is active at all times can reduce following error on an extended profiled move, but at the cost of reduced system stability and of overshoot at the end of the move (which makes up for the lag at the beginning of the move). In a system without feedforward, the close following may be worth these costs. But the velocity and acceleration feedforward terms in PMAC can virtually eliminate

following error without these drawbacks. For this reason, most PMAC customers use error integration only when motion is stopped – where it can eliminate steady state errors due to static friction or net torque loads. Proportional gain term Kp (Ix30) also affects derivative and integral gains, and is not strictly speaking a true proportional gain. For this reason, if you modify Kp when Ki (Ix33) or Kd (Ix31) is not equal to zero, you are also changing the effective integral or differential gain. The shape of the response curve will not change much, although its timing will. You will want to change Ki and/or Kd in the opposite direction from Kp if you want to keep their effective gains constant.

Feedforward will affect step response even though it has no effect on the system stability we are really evaluating. Be sure that both acceleration and velocity feedforward are set to zero as you are doing the step responses.

The default step size of 100 encoder counts may or may not be adequate. The guidelines are to make the step large enough so that the granularity of the position measurement is not a nuisance, but small enough so that the filter does not saturate on the step (the step size times proportional gain should be less than 178,950,000 with some margin; for instance a step size of 3000 with a proportional gain of 60000 will saturate, giving a misleading response).

Some systems will have mechanical resonances in the coupling of the motor to the load. The PID filter cannot compensate for these resonance's; if their presence is not tolerable, you must keep the gains low enough not to stimulate them, or (preferably) stiffen the coupling to reduce the resonances.

Doing the Step Response:

1. Set a safe starting filter with a little proportional gain, with no (or almost no) derivative or integral gain, and no feedforward. The current values for Kp, Ki, and Kd are displayed on the screen.
2. Do a step move and observe the plotted response displayed on the screen, along with the calculated statistics.
3. Adjust (probably increase) Kp (Proportional Gn) to get the fastest rise time possible without a huge amount of overshoot. Allow more overshoot here than you will in your final response.
4. Once you have a fast response, increase Kd (Derivative Gn) to bring down the overshoot to the desired value. Note that this will also increase the rise time.

You may need to do further tradeoffs between Kp and Kd to get the desired response.

Note:

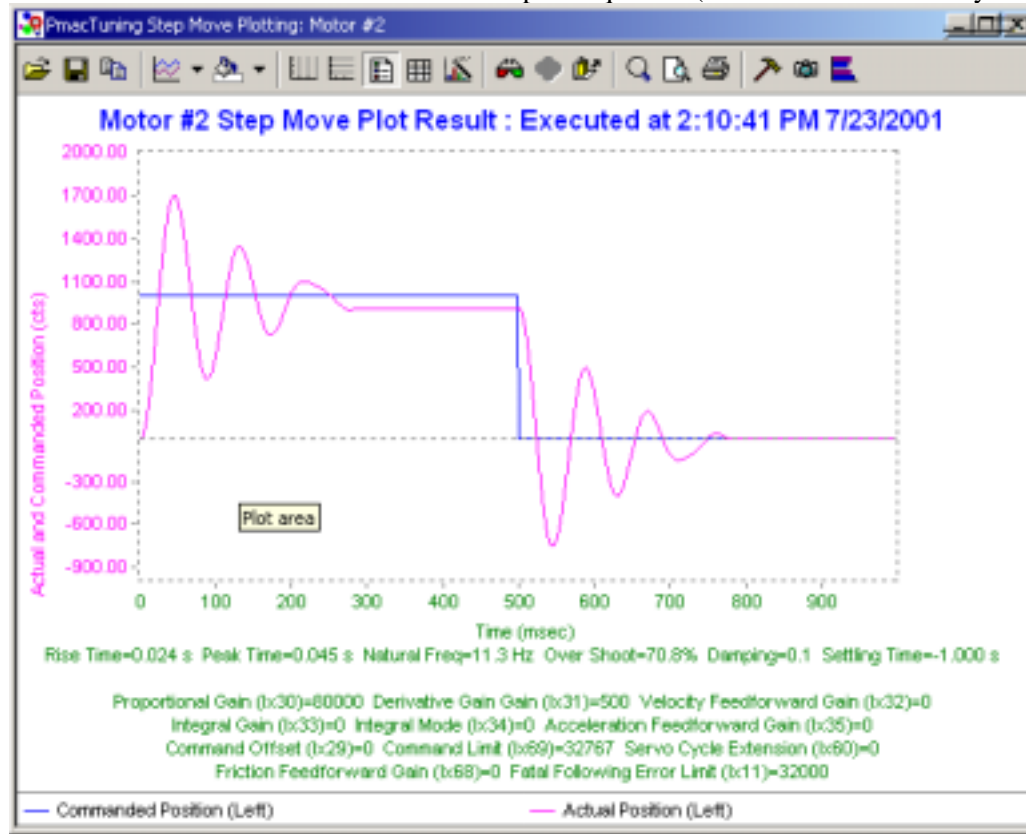
You may wish to change the size and/or the duration of the step to be able to observe the response better. The default values are 1000 counts step with 500 milliseconds dwell time.

5. Once you have set Kp and Kd, you have taken care of your dynamic step response (provided you are using error integration only in position). Now you will want to add integration to improve the static holding properties of the system. As you increase Ki (Integral Gn) and observe the step response, you will notice that it increases overshoot but comes back to the command position more quickly. A good value for Ki is one that brings the response back down to the command position as quickly as possible without going back past it.

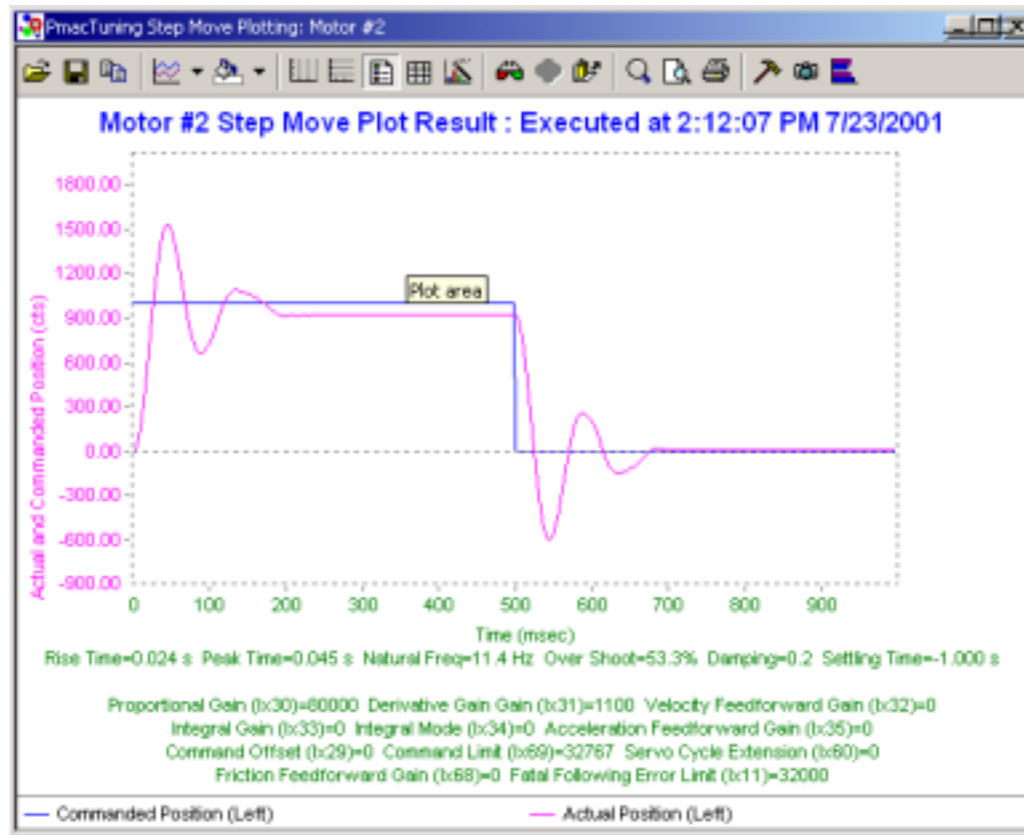
The following figures are taken from the actual tuning procedure for a motor using this program. They were done on a small DC laboratory motor with virtually no load, so the response is faster than it would be in almost any real-world application. The actual times are not important. However, the shapes of the response curves are. This system has a current-controlled amplifier with no tachometer. The goals of this system are critically damped step response, the quick elimination of steady-state errors at rest, and the minimization of following errors.

Initial step move response.

The screen below shows the step response with very little damping added (derivative gain $K_d = 500$). The large amount of ringing is unacceptable; therefore some more K_d is needed. The next figure shows the response with an increased K_d of 1100. The ringing is largely eliminated; there is an overshoot of 54% and a rise time of 24 msec. Let us see if we can make the response quicker (which means a stiffer system).

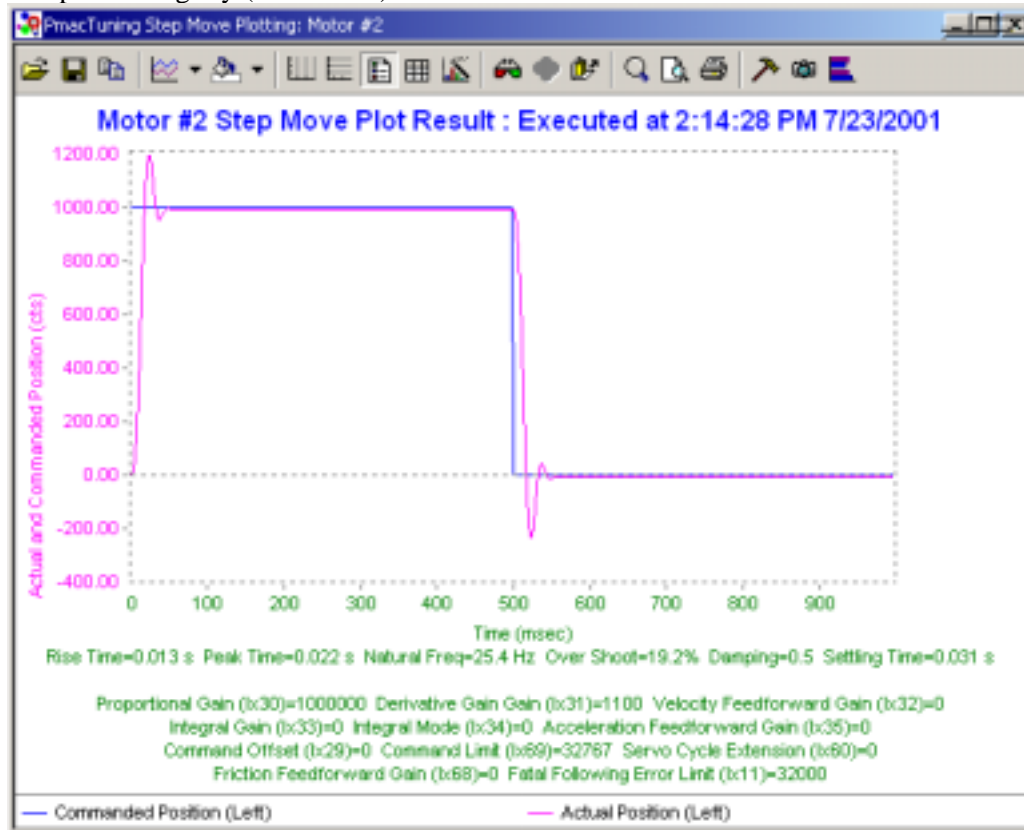


Modified step response with higher Kd (Ix31).



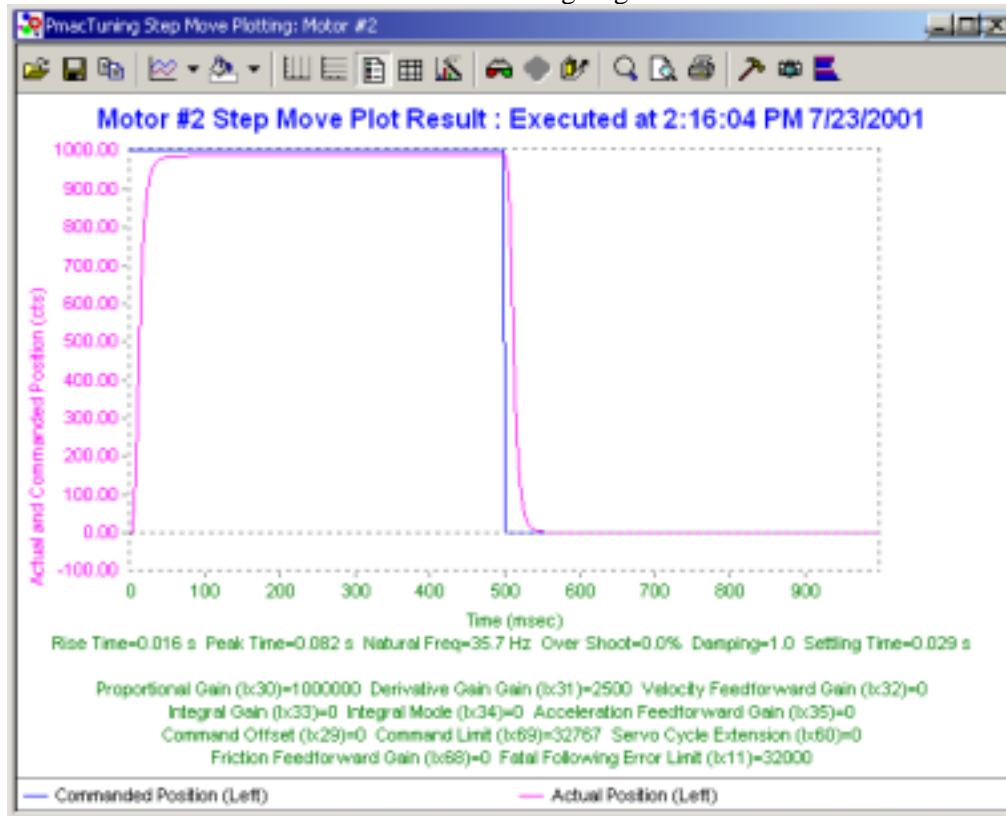
Modified step response with higher Kp (Ix30).

The next figure below shows the response with Kp increased from 80000 to 1000000. Note that the shape of the curve has not changed much (this is because the effective derivative gain is increasing with Kp), but the rise time has improved slightly (to 13 msec). We turn now to Kd.



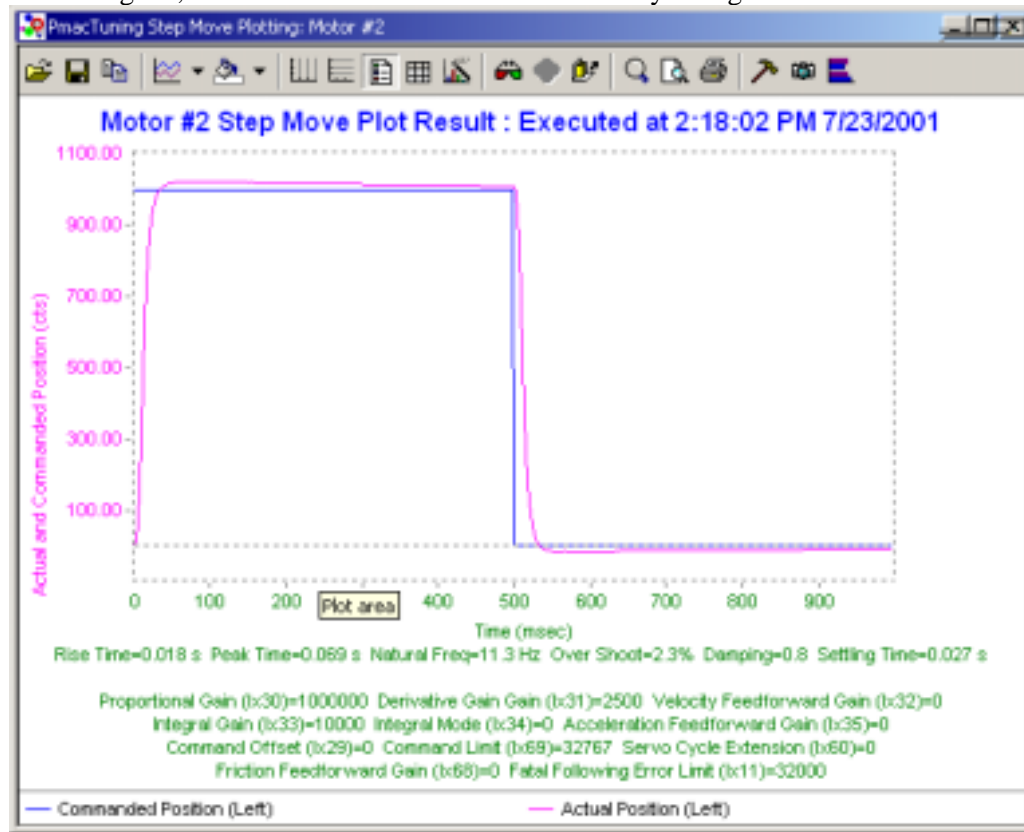
Modified step response with increased Kd again.

The figure below shows the step response with a Kd of 2500. This is the critically damped case; i.e. fast response with no overshoot. With Kd any smaller, we get some overshoot. And with it any larger, we just slow down the response. The tendency of the system to settle slightly off from the target position is due to a net torque or static friction. We will eliminate this with integral gain.



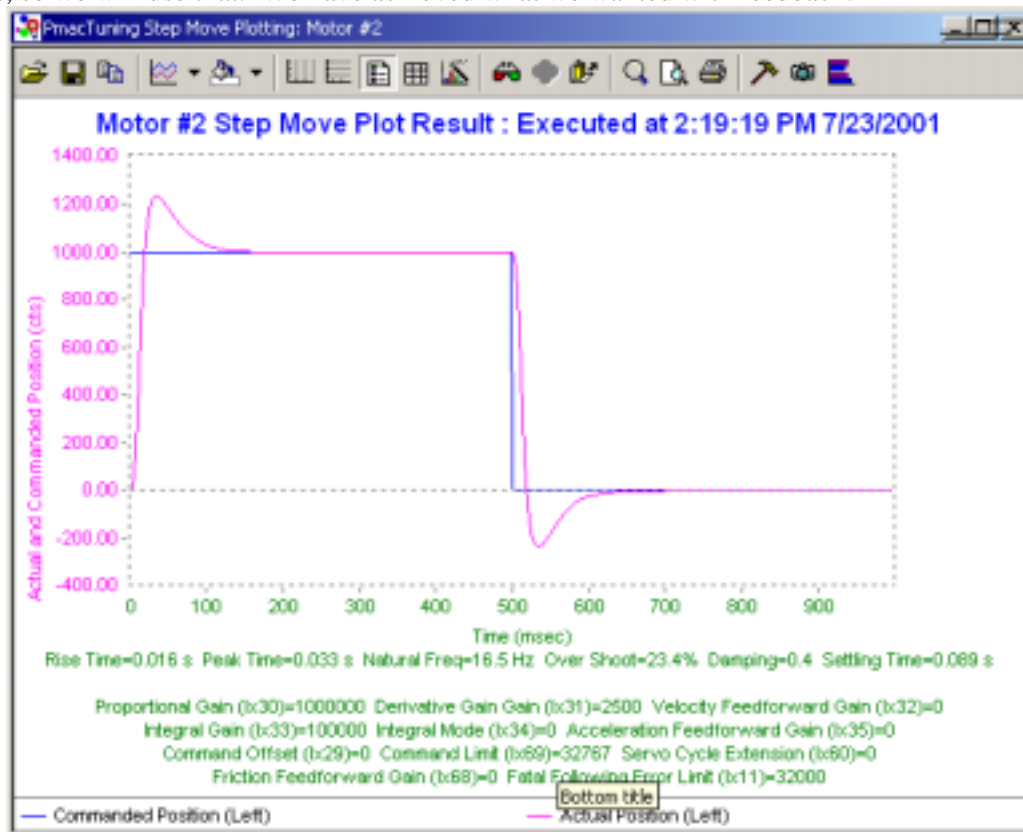
Modified step response with increased Ki (Ix33).

The figure below shows what happens with a little bit (relatively speaking) of integral gain ($K_i = 10000$): the steady-state error is gone, but the nature of the curve has not really changed.



Modified step response with very high Ki.

The figure below shows the response curve for a substantially increased Ki (100000). This curve demonstrates how quickly the system would respond to a disturbance while in position. (Remember that we are using integral gain only when in position, so changing integral gain does not affect our actual dynamic response, although it will change the shape of the step response here. We still have the stability characteristics of critical damping while on the move.) Even the higher value of Ki does not result in oscillations, so we will use that. We have achieved what we wanted with feedback.



Position Ramp

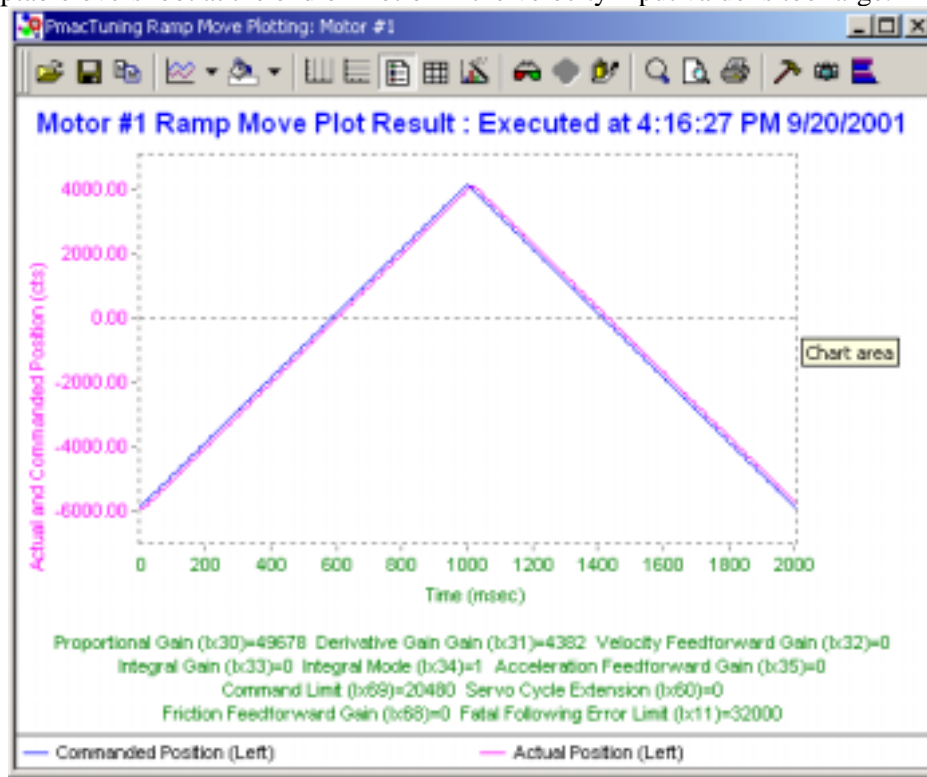
The Ramp Position input test profile provides a forward/reverse trajectory with almost zero acceleration time (very large acceleration rate) and a constant velocity.

Position Ramp
Move Distance (cts):
Velocity (cts/sec):
Number of Repeats:

Do A Ramp Move

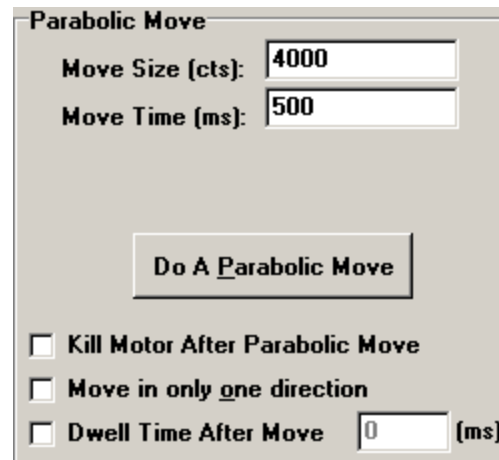
☐ Kill Motor After Ramp Move
☐ Move in only one direction
☐ Dwell Time After Move (ms)

This trajectory is suited best for the tuning of the Velocity Feedforward gain (Ix32) in order to reduce the steady state following errors at a constant velocity. Note that a too high value for the velocity input may cause unacceptable mechanical stress – even vibrations on the drive system and its load. In addition, there may be unacceptable overshoot at the end of motion if the velocity input value is too large.



Parabolic Velocity

(Feedforward Tuning with a Profiled Parabolic Response)



In a position servo system without feedforward or dynamic error integration, there must be a continual error between the commanded position and the actual position in a profiled move (known as following error) to produce a motor command. This means, however, that you are never really where you want to be when you want to be there, which is often the point of a profiled move in the first place. These following errors will usually be proportional (well correlated) to the velocity and/or the acceleration. The velocity and acceleration feedforward terms can be used to reduce these following errors virtually to zero. These parameters add terms to the torque command that are proportional to the commanded velocity and acceleration, respectively, in each cycle of the profiled move.

Mathematically speaking, if two sets of data, such as velocity and following error, vary in complete proportion to each other, they have a correlation of 1.0 (perfect correlation). If they vary completely independently of each other, they have a correlation of 0.0 (no correlation). The more they vary in proportion to each other, the closer their correlation will be to 1.0. In graphical terms, the more two curves are shaped like each other, the better they will be correlated. Another important figure is the constant of proportionality between the two sets of data, which is the average ratio between matching points in the sets. Even if two sets of data are very well correlated, the ratio may be so low that one of the sets is negligible in practical terms. For each move done, the program will calculate the correlation between velocity and following error, and between acceleration and following error. It will also calculate the average ratio between following error and both velocity and acceleration. As a feedforward gain is increased, the ratio of following error to that quantity will decrease linearly (e.g. if a gain of 0 produces a ratio of 12.0, and a gain of 5000 produces a ratio of 6.0, then a gain of 10,000 can be expected to drive the ratio to zero). The ratio will decline even as the correlation stays high. When the ratio gets small enough, the correlation should decrease as some other factor becomes the dominant cause of following error (e.g. noise or the other feedforward gain). Ideally, the correlation will be brought near zero as well as the ratio.

PMAC has two feedforward terms:

- Kvff: Velocity Feedforward gain (Ix32)
- Kaff: Acceleration Feedforward gain (Ix35)

The strategy in this section is to do a series of "parabolic" moves (cubic in position, parabolic in velocity, linearly varying in acceleration), while adjusting the feedforward terms to reduce the following error and its correlation to velocity and acceleration. After each move, the program will automatically calculate the correlations and ratios, and the maximum following error. These will be displayed on a plot of the position and the following error. The feedforward terms are increased from zero (working first with velocity feedforward) until the ratios (and hopefully the correlations) are as close to zero as possible, without going

strongly negative. If either correlation goes very far negative, you will be likely to get overshoot at the end of a move.

To get meaningful correlation information, particularly about acceleration, you must push the system hard. By increasing the length and/or decreasing the time of the move, you can get higher velocities and accelerations. Decreasing time is appropriate if increasing the length would cause problems with maximum travel or top velocity.

In a system without a tachometer, you will probably want to set the velocity feedforward equal to the derivative gain or slightly greater. In a system using a tachometer, the velocity feedforward should be greater than the derivative gain.

When you do the parabolic move without any feedforward, you will probably see a very high correlation to velocity (≈ 1.0) and almost no correlation to acceleration (≈ 0.0). There is most likely a real correlation to acceleration, but it is swamped out by the velocity correlation. As you reduce the velocity correlation, you should see increased acceleration correlation. When you then reduce the acceleration correlation, the correlation to velocity may increase again, but the actual ratio and magnitude of the following error should be very small.

Parabolic moves were chosen for this procedure because their acceleration and velocity vary continually and are uncorrected to each other, making them ideal for this type of analysis. For further examination of the move, you may plot the velocity curve, the acceleration curve, or the following error curve. These are automatically scaled to fill up the plot window for maximum resolution. The move statistics are re-displayed with these plots.

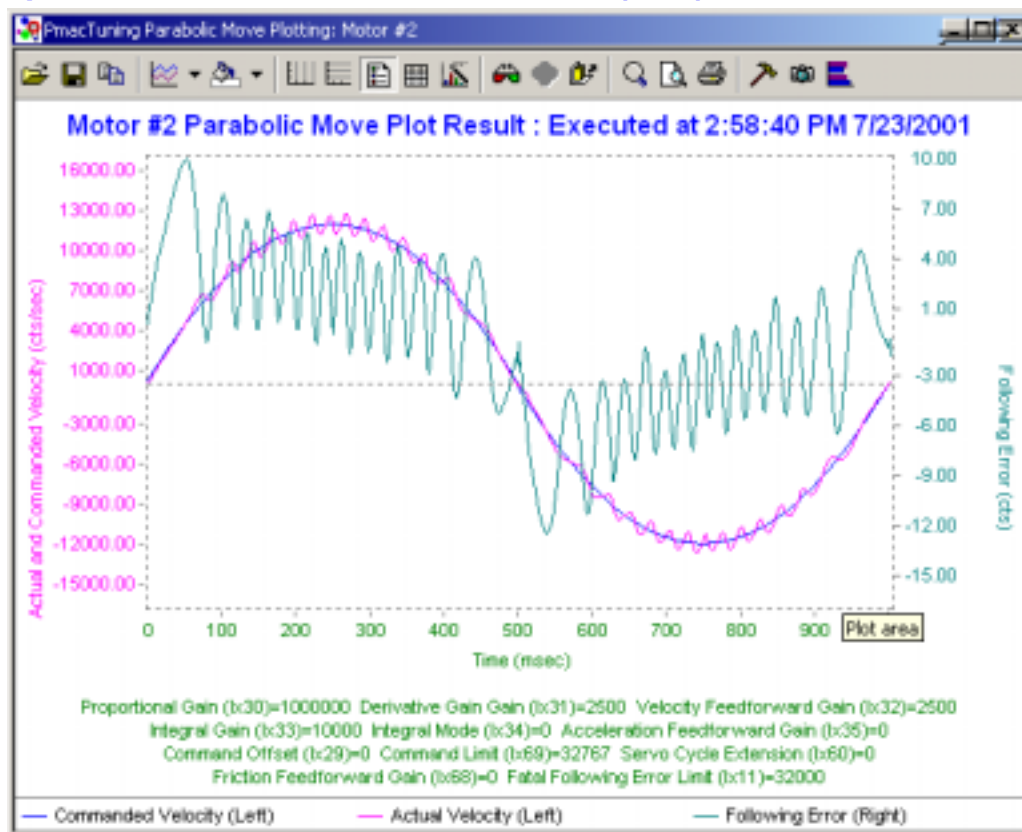
Doing the Parabolic Move:

1. Do a parabolic move and observe the plotted response and the calculated statistics for the move.
2. Presuming there is a high correlation between following error and velocity (the velocity and following error curves have the same shape) and a relatively high maximum following error (as shown on the plot), increase Kvff (Velocity FF Gn Ix31).
3. Do another parabolic move. If there is still inadequate Kvff, there will still be a high correlation, but the FE-to-Vel ratio and the maximum FE will be reduced. Repeat steps 2 and 3 until you have the maximum Kvff that produces a square wave looking shape for the following error.
4. At this point, there should be noticeable correlation between acceleration and following error. You can see this by the numerical correlation value, or by plotting the acceleration and the following error, and noticing the similarity in shape. If you do not see any correlation, try increasing the length or decreasing the time of the move to get higher accelerations.
5. Increase Kaff (Accel FF Gn Ix35). Do another move, and evaluate the correlation and FE-to-Accel ratio again. Repeat until you have the maximum Kaff that retains any positive correlation. At this point, you should have minimal following error, and most of what remains should be caused by noise or mechanical friction. If mechanical friction is the cause you will see a fairly strong velocity correlation because the friction causes the following error related to the sign of the velocity. However,

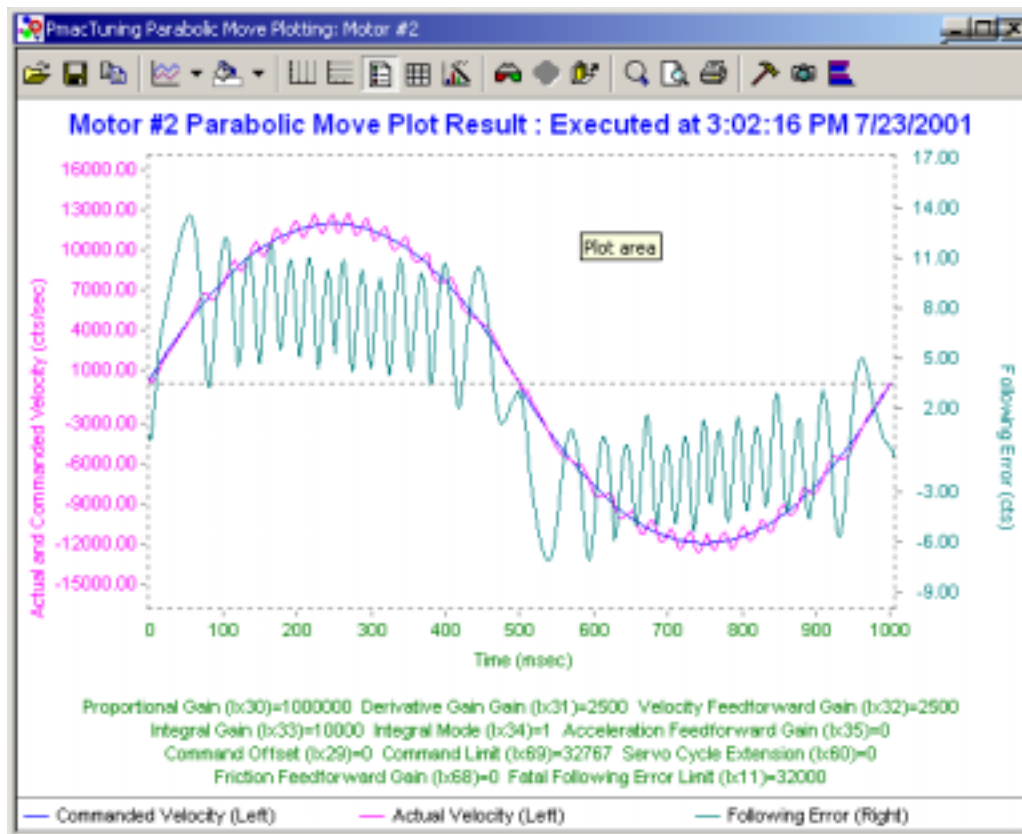
each half of the following error curve should be quite flat if friction is the primary cause.

In the first figure below, we see the plot of a parabolic move using a step size of 4000 counts and a step time of 500 milliseconds. Here, position is plotted against time, with the statistics of the following error shown. The following error gets as large as 112 counts, and it is almost perfectly correlated to velocity (Vel corr = 0.913). It shows very low correlation to acceleration (Acc corr = 0.226). The next two figures show the plot of position, and velocity and following error, versus time for the same move. Note that the velocity and following error curves are shaped almost exactly alike. This is a ramification of their high correlation. Now let's introduce some velocity feed forward. As a rule of thumb, choose Kvff (Ix32) to be equal to Kd (Ix31). In this case, Kvff = 2500. By looking at the following error curve, it no longer resembles the velocity curve (correlation of 0.211). Also, the maximum following error is reduced from 112 to 9.6 counts!

Modified parabolic move with increased Kvff (Ix32).

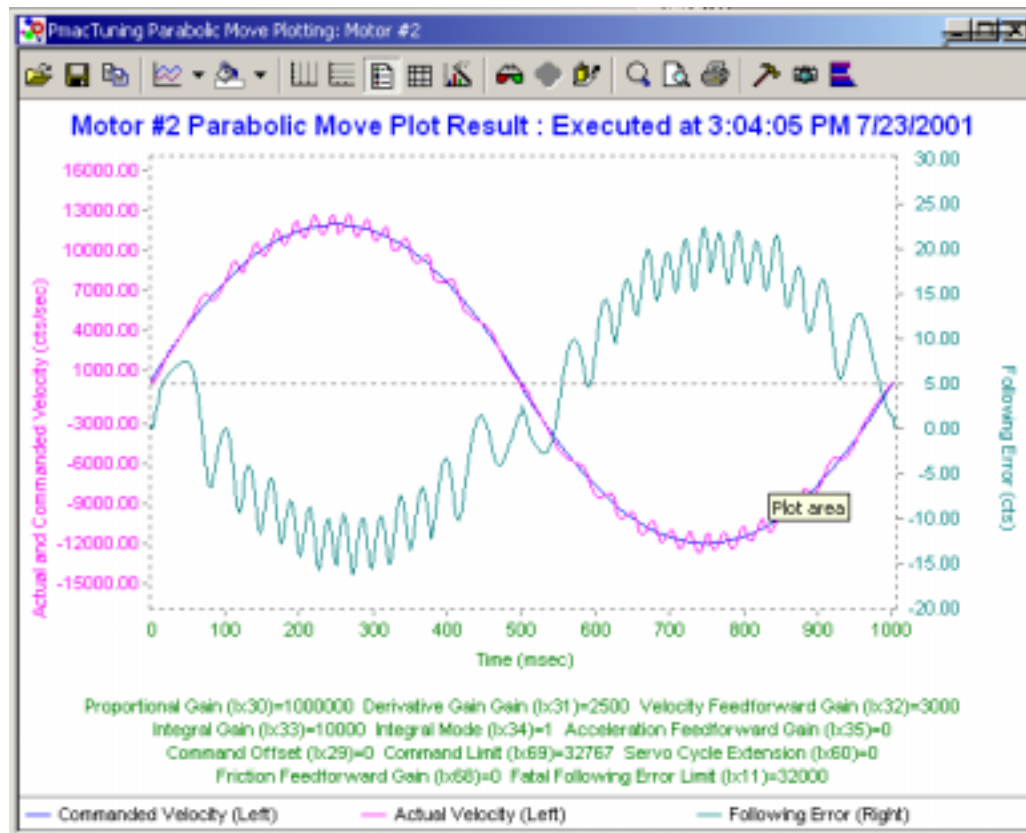


Modified parabolic move with Ix34 = 1.

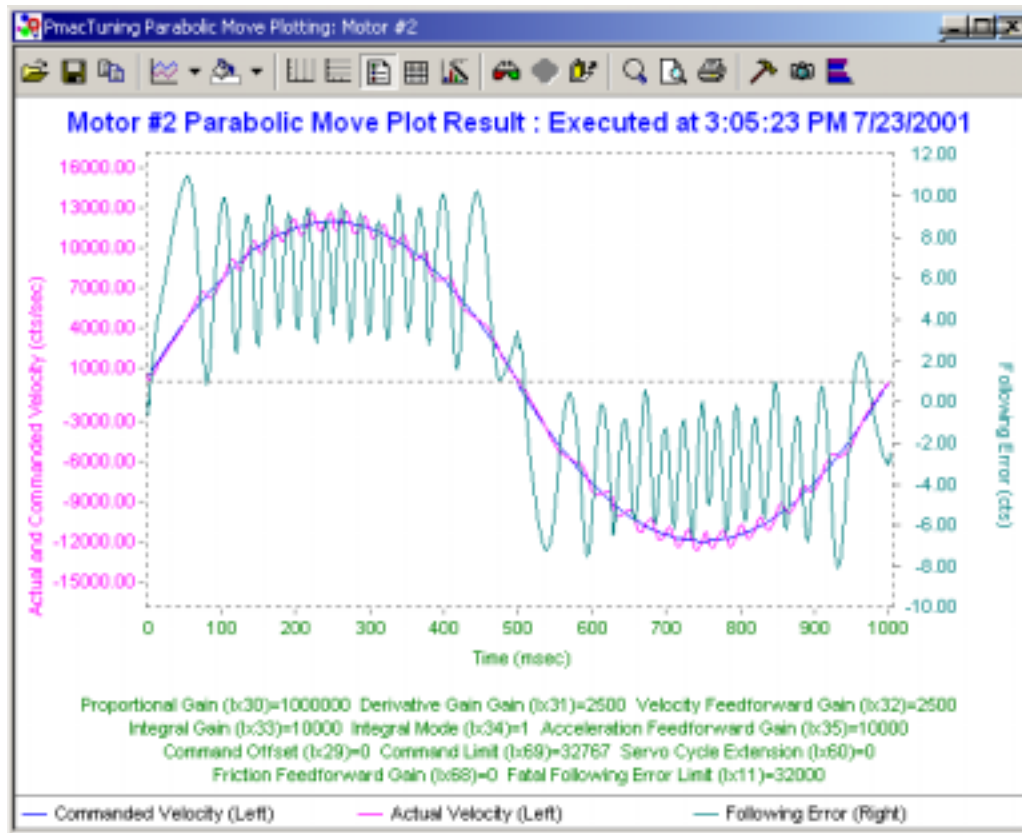


Now we instruct the integration in the servo loop to be active only when the motor is not be commanded to move. To do this, set Ix34, the integration mode, equal to 1. The improvement is seen above. If we continue to increase Kvff, then we have the following behavior where the following again resembles the velocity curve, but in an inverted manner.

Modified parabolic move with too much Kvff.



Modified parabolic move with some Kaff.



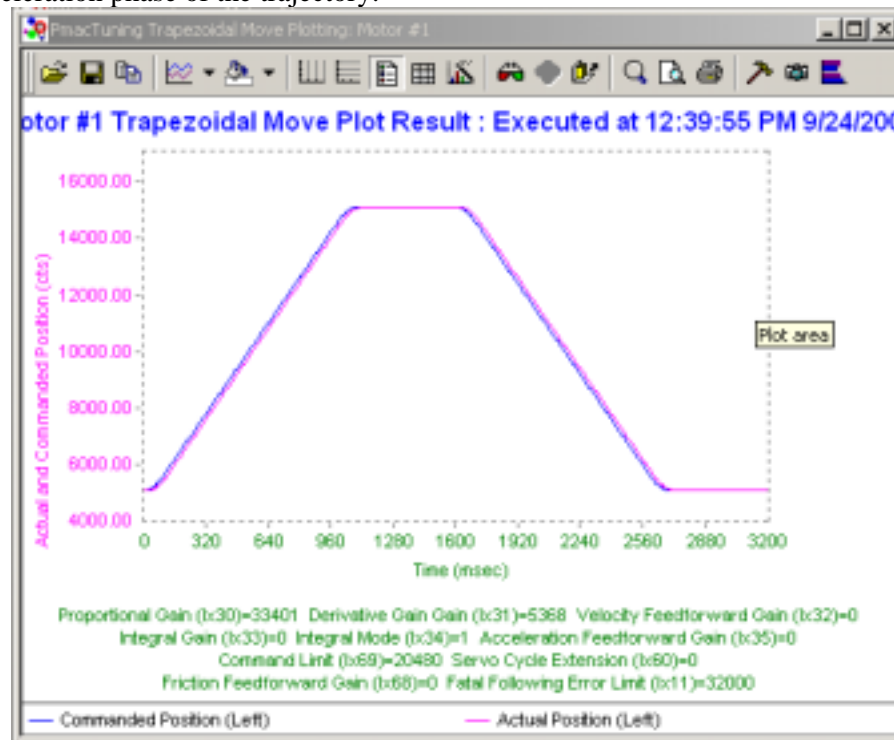
Finally, a bit of acceleration feedforward, Kaff (Ix35), gives us an almost perfectly tuned motor!

Trapezoidal Velocity

Trapezoidal Velocity	
Move Distance (cts):	10000
Velocity (cts/sec):	10000
Acceleration (cts/msec):	100
Number of Repeats:	1
<input type="button" value="Do A Trapezoidal Move"/>	
<input type="checkbox"/> Kill Motor After Trapezoidal Move	
<input type="checkbox"/> Move in only one direction	
<input type="checkbox"/> Dwell Time After Move	<input type="text" value="500"/> (ms)

The Trapezoidal Velocity input test profile provides a forward/reverse trajectory with a constant acceleration rate. You select a desired acceleration time as well as the peak velocity. If the peak velocity could not be reached (due to the indirectly specified acceleration rate) then a triangular velocity profile will be generated.

After the Velocity Feedforward gain has been fine tuned with either the Parabolic Input or the Ramp input, you may use this trajectory in order to fine tune the Acceleration Feedforward gain (Ix35) during the acceleration/deceleration phase of the trajectory.



S-Curve Velocity

S-Curve Velocity

Move Distance (cts):

Velocity (cts/sec):

Acceleration (cts/msec):

Number of Repeats:

☐ Kill Motor After S-Curve Move

☐ Move in only one direction

☐ Dwell Time After Move (ms)

The S-curve Velocity input test profile provides a forward/reverse trajectory with a constant rate of change of acceleration or constant jerk. You select a desired acceleration time as well as the peak velocity. If the peak velocity could not be reached (due to the indirectly specified acceleration rate) then a parabolic velocity profile will be generated.

After the Velocity Feedforward gain has been fine tuned with either the Parabolic Input or the Ramp input, you may use this trajectory in order to fine tune the Acceleration Feedforward gain (Ix35) during the

acceleration/deceleration phase of the trajectory. Note that the S-curve Velocity input does provide a controlled rate of change of acceleration/deceleration which, in turn, reduces the mechanical stress on the drive system. However, for the same acceleration time and final speed, the peak value of acceleration will be twice the Trapezoidal input.

Sinusoidal

CAUTION

Depending on the controller gain, for even small amplitude of the sine wave, the DAC output under closed loop control may oscillate at its maximum magnitude during sine trajectory particularly at higher frequencies. This may cause motor, amplifier and/or mechanical structure damage. ***Do not execute a sinusoidal trajectory with a high frequency if you are uncertain about the ability of your electromechanical hardware to withstand the current or torque demands generated by the high frequency sinusoidal input.***

The image shows a software dialog box titled "Sinusoidal". It has three input fields: "Amplitude (cts)" with the value "1000", "Frequency (Hz)" with the value "1", and "Number of Repeats" with the value "10". Below these fields is a button labeled "Do A Sinusoidal Move". At the bottom of the dialog is a checkbox labeled "Kill Motor After Sinusoidal Move" which is currently unchecked.

The **Sinusoidal** dialog box allows the entry of commanded position as a sine wave of a given frequency. The option is very useful for testing the frequency response of the closed loop system at a particular frequency. The **Amplitude size** is incremental from the present command position. The **Frequency** is in units of Hz and you specify the number of sinusoidal cycles.

Sine Sweep

CAUTION

Depending on the controller gain, for even small amplitude of the sine wave, the DAC output under closed loop control may oscillate at its maximum magnitude during the sine sweep, particularly at higher frequencies. This may cause motor, amplifier and/or mechanical structure damage. ***Do not execute a sine sweep trajectory with a high max. frequency if you are uncertain about the ability of your electromechanical hardware to withstand the current or torque demands generated by the sine sweep at high frequencies!.***

Sine Sweep

Amplitude (cts): 1000

Start Frequency (Hz): 1

End Frequency (Hz): 10

Sweep Time(sec): 10

Do A Sine Sweep Move

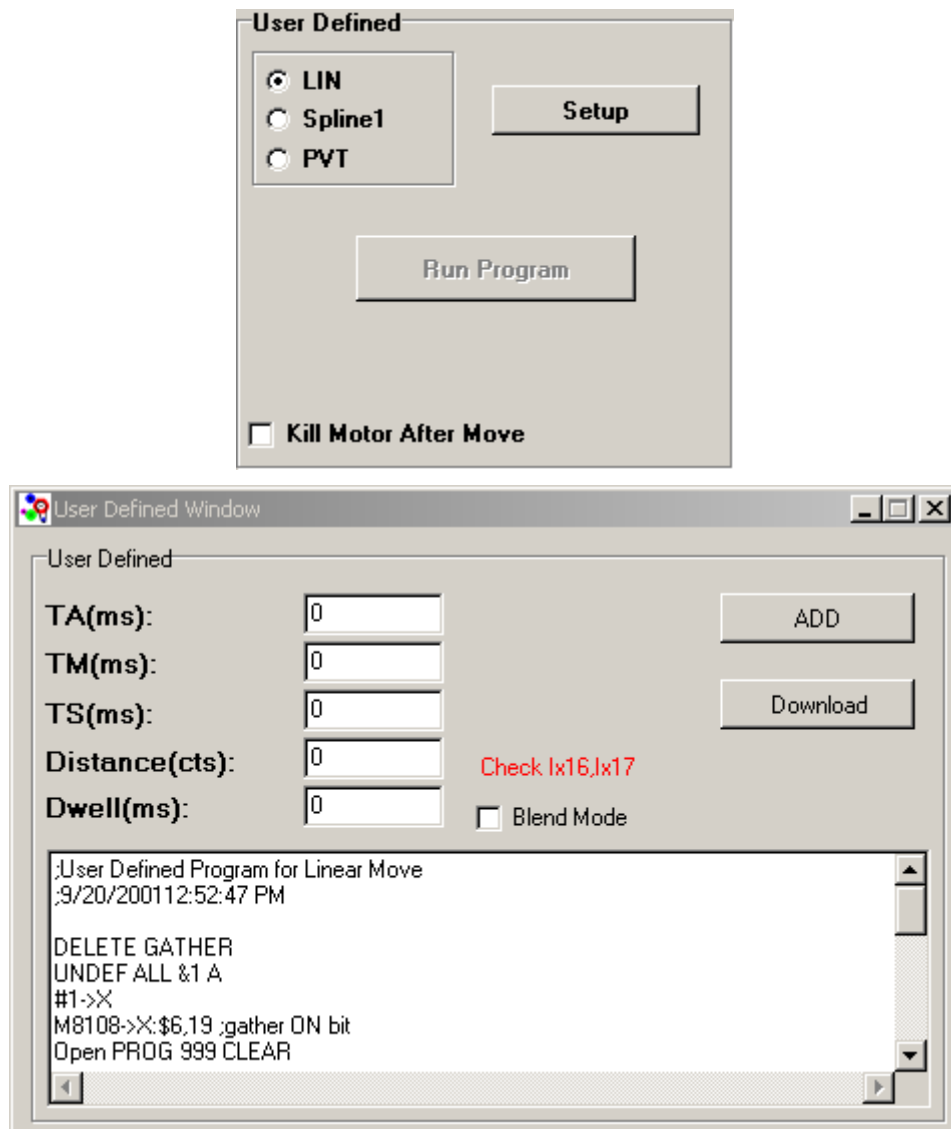
☐ Kill Motor After Sinusoidal Move

Sweep Method: ☐ Logarithmic ☒ Linear

Domain: ☒ Time ☐ Frequency

The **Sine Sweep** dialog box allows the entry of commanded position as a sine wave of varying frequency. This option is useful for testing the frequency response of the closed loop system through a range of frequencies. From the resulting sine sweep a *bode plot* may be obtained within the **Plotting** menu. The entries include the **Amplitude** of sinusoidal input, the **Starting and Ending Frequencies** in Hz, the **sweep time**. In addition, whether a linear or logarithmic method should be used for the frequency increase. The data gathering parameter 119 is automatically adjusted so that the highest frequency is sampled at least eight times. As a result a long sweep time should be avoided when a wide frequency range is chosen that ends at a high frequency (since the number of data gathering storage area within PMAC is limited to approx. 10000 points). The *linear* sweep changes the frequency from the minimum to maximum as a linear function of time, which provides more cycles at higher frequencies. The *logarithmic* sweep changes the frequency from the minimum to the maximum, as log10 function of time provides more cycles at lower frequencies. In theory, the linear should lead to more accurate high frequency bode plots, and the logarithmic to more accurate low frequency bode plots.

User Defined

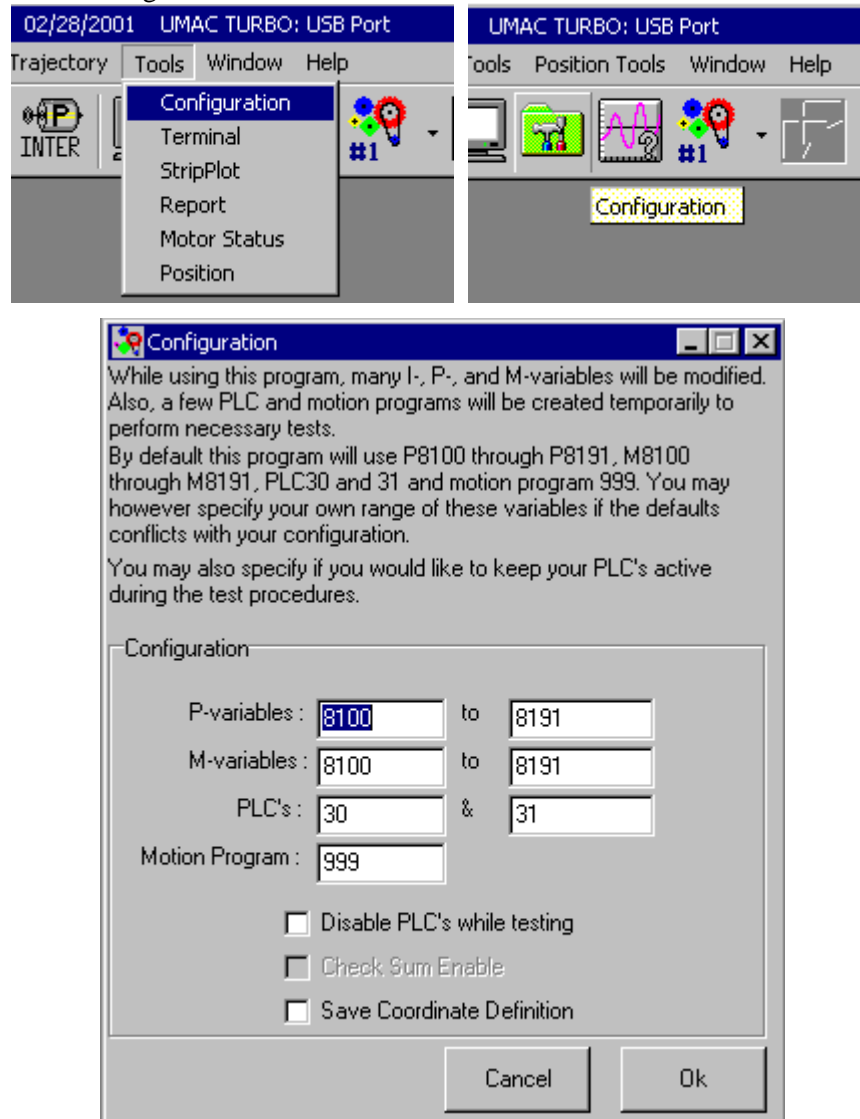


The User Defined input test profile provides a forward/reverse trajectory with Linear Mode, Spline1 Mode, and PVT Mode. You select a desired TA, TM, TS Distance and Dwell Time and repetition. After adding movement, you can see all motion programs code in the EditBox. You can select the blend mode or not, then download the motion program to pmac and run from the trajectory window.

CHAPTER 6 - TOOLS

Configuration

This command displays Configuration Windows. You can set P-variables range, M-variables range, PLC Number, Motion Program Number to use Pmac Tuning Program, and enable other PLCs or disable PLCs, when tuning with PmacTuningPro.



P-variables: You can set at least 50 variables between Start P-variables and End P-variables.

M-variables: You can set at least 50 variables between Start M-variables and End M-variables.

PLC number: You can set 2 legal plc numbers to use in PmacTuningPro.

Motion Program: You can set 1 legal motion program number to use in PmacTuningPro.

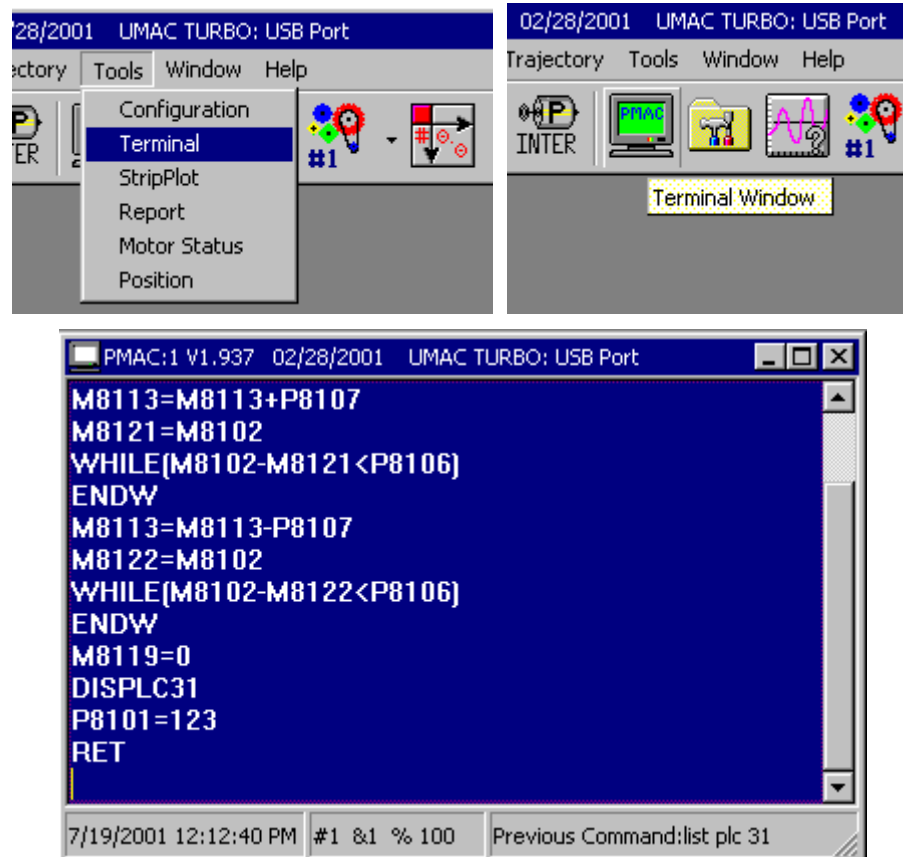
Disable PLCs while testing Checkbox: You can enable or disable any other currently running PLCs whole doing auto tune or trajectories.

Check Sum Enable Checkbox: You can set enable or disable when you upload data using check sum or not in serial communication mode.

Save Coordinate Definition Checkbox: You can set enable or disable when you do trajectories to save your coordinate definition and restore your coordinate definition after done trajectories.

Terminal

This button calls a terminal window that can be used to send on-line commands and to query variables from PMAC.



You can send any command or see return value by return key.

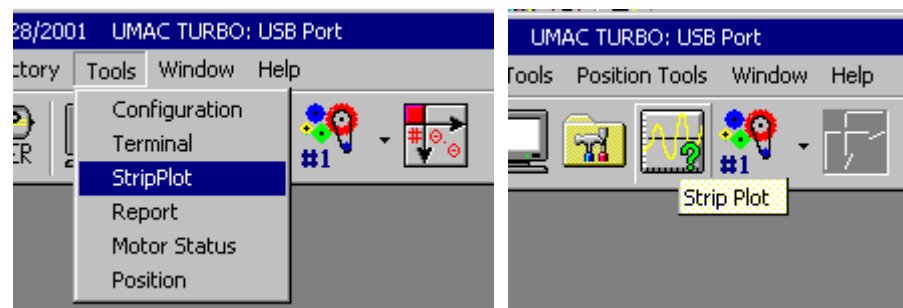
You can see the current motor number, coordinate number and feedrate override value.

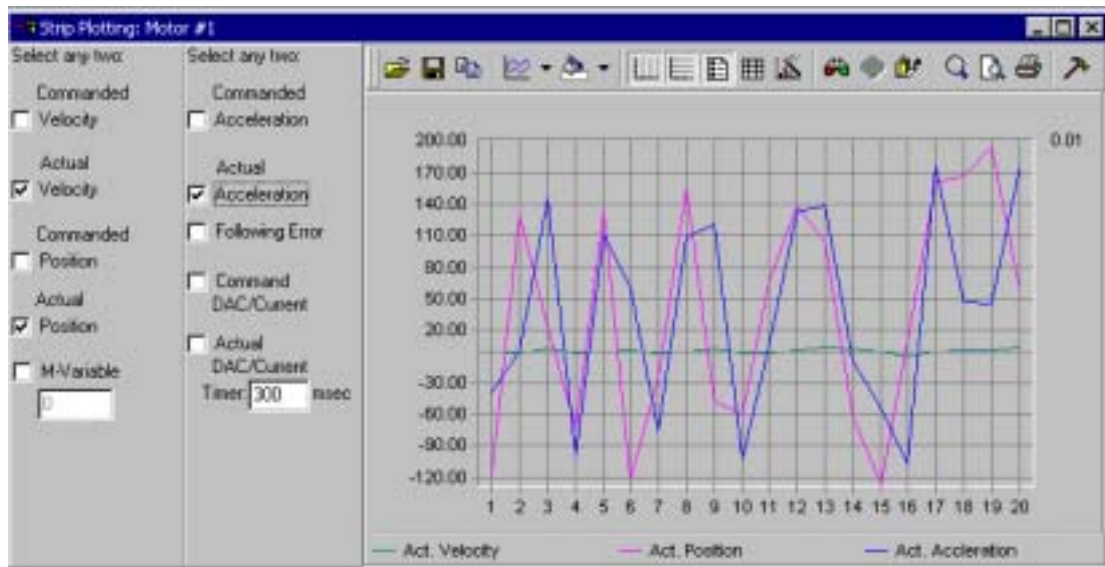
You can use up-arrow or down-arrow to catch the previous command history.

You can set Font or Background setting by clicking Terminal Tools from main menu.

Strip Plot

This button calls a Strip Plot window that can be used to see real time data from PMAC.





Commanded Velocity: You can see the commanded velocity from PMAC in the left side plot.

Actual Velocity: You can see the actual velocity value from PMAC in the left side plot.

Commanded Position: You can see the commanded position value from PMAC in the left side plot.

Actual Position: You can see the actual position value from PMAC in the left side plot.

M-Variable: You can see the user defined M-variable value from PMAC in the left side plot.

Commanded Acceleration: You can see the commanded acceleration value from PMAC in the right side plot.

Actual Acceleration: You can see the actual acceleration value from PMAC in the right side plot.

Following Error: You can see the following error value from PMAC in the right side plot.

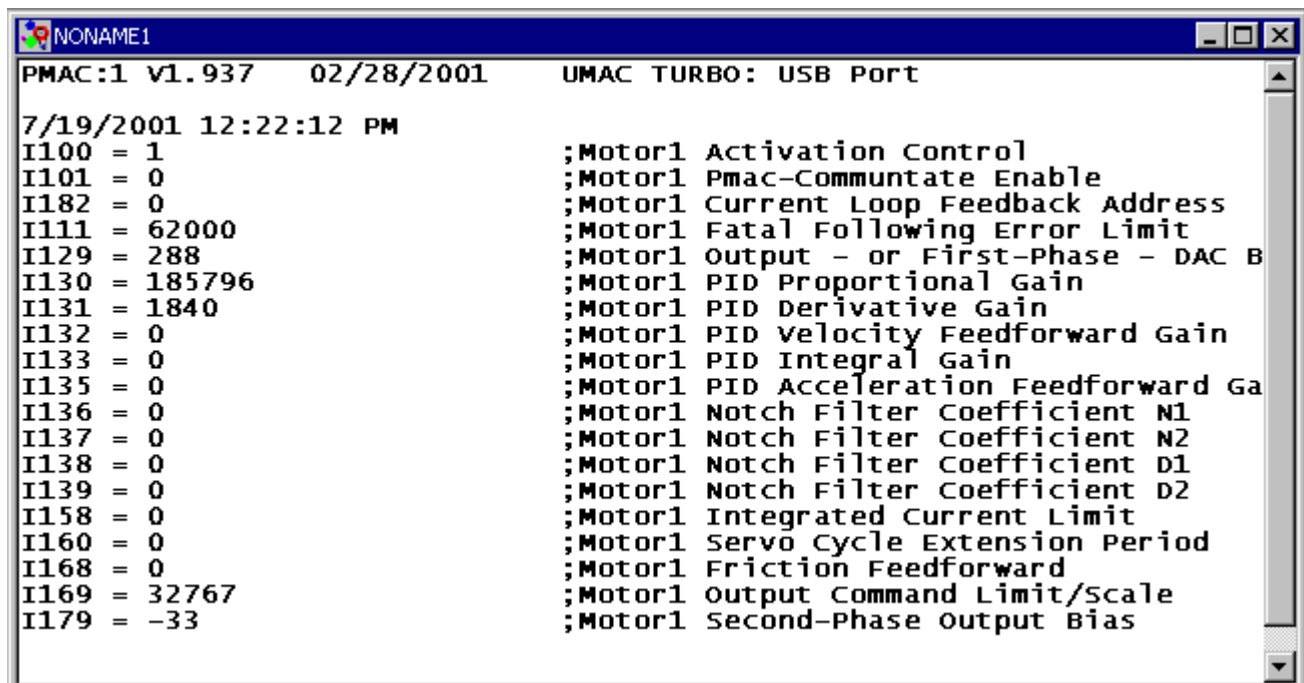
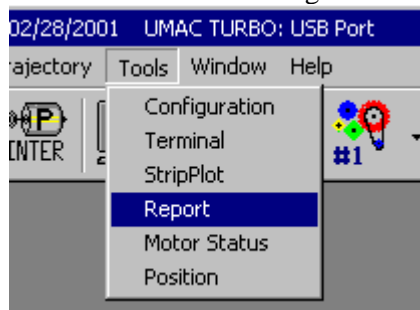
Command DAC/Current: You can see the command DAC/Current value from PMAC in the right side plot.

Actual DAC/Current: You can see the actual DAC/Current value from PMAC in the right side plot.

Timer: You can set the data update time interval.

Report

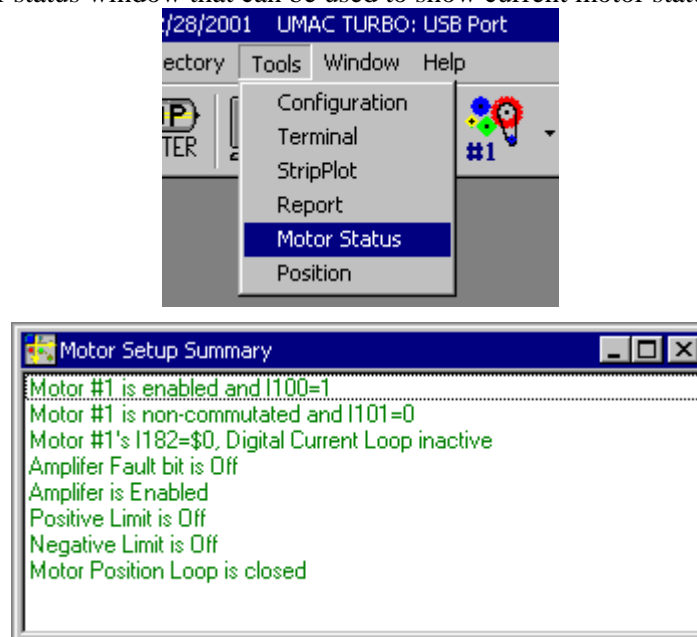
This button calls a report window that can be used to see tuning related I-variables.



You can see the I-variables that relate PMAC tune values and print out the report or save it to file.

Motor Status

This button calls a motor status window that can be used to show current motor status in real time.

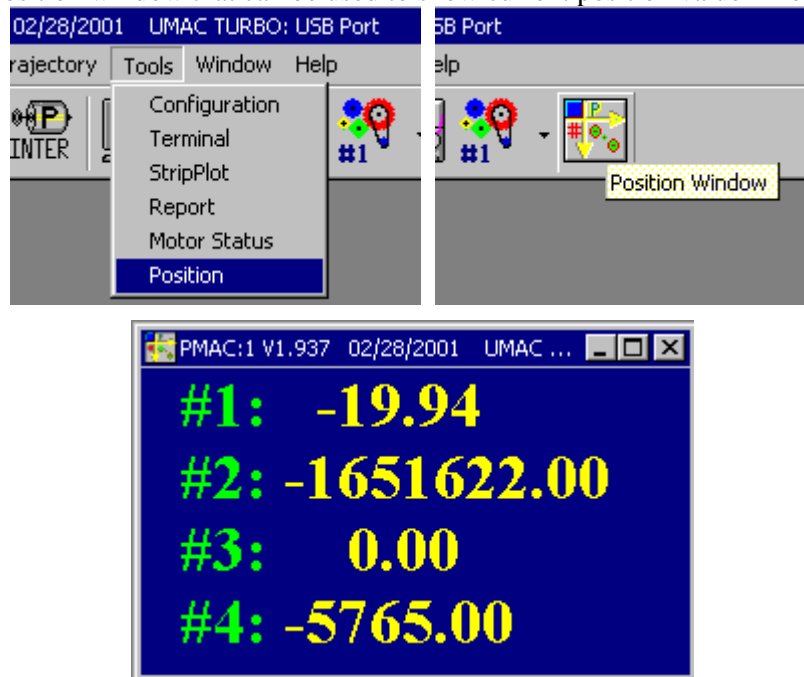


You can see the current motor status in real time:

- Current motor activated or not. (Ix00)
- Current motor commutation enable or not. (Ix01)
- Current motor current loop feedback address. (Ix82 : PMAC2 type only)
- A zero value for Ix82 tells PMAC2 not to close the current loop for this motor.
- Current motor amplifier fault bit is on or off.
- Current motor amplifier enable bit is on or off.
- Current motor positive limit bit is on or off.
- Current motor negative limit bit is on or off.
- Current motor position loop mode is killed, opened, or closed.

Position

This button calls a position window that can be used to show current position value in real time.



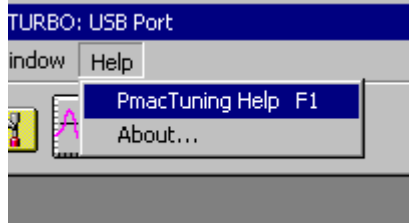
You can select which motor position value to display by clicking the right-mouse button and **open select motor** command.

You can set Font, Size, Color and Background by clicking right-mouse button and **open** command.

CHAPTER 7 - HELP

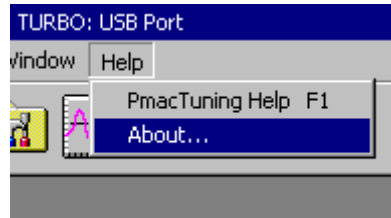
Pmac Tuning Help

This button calls a Help window that can be used to help using PmacTuning Program.



About

This button calls an "About" window that can be used to show version of PmacTuningPro and copyright.



You can see Version and created date.

You can see Delta Tau Data Systems, Inc. web site and copywrite.